Calcul de pose dynamique avec les caméras CMOS utilisant une acquisition séquentielle

Ludovic Magerand

N°d'ordre : DU 2534
EDSPIC : 681

Université Blaise Pascal - Clermont II

Ecole Doctorale Sciences Pour l'Ingénieur de Clermont-Ferrand

Thèse

Présentée par

LUDOVIC MAGERAND

En vue de l'obtention du grade de

Docteur d'Université

Spécialité Vision par Ordinateur

Calcul de pose dynamique avec les caméras CMOS utilisant une acquisition séquentielle

Soutenue publiquement le 18 décembre 2014 devant le jury

Jean-Marc Lavest	Prof, Université d'Auvergne	Président du jury
David Fofi	Prof, Université de Bourgogne	Rapporteur
Thomas Corpetti	DR, Université Rennes 2	Rapporteur
Nicolas Andreff	Prof, Université Franche-Comté	Examinateur
Daniel Pizarro	Dr, Université d'Auvergne	Invité
Omar Ait-Aider	Dr, Université Blaise Pascal	Co-Encadrant
Adrien Bartoli	Prof, Université d'Auvergne	Directeur de thèse

Préparée à

Institut Pascal – Institut des Sciences de l'Image pour les Techniques interventionnelles

RÉSUMÉ

En informatique, la vision par ordinateur s'attache à extraire de l'information à partir de caméras. Les capteurs de celles-ci peuvent être produits avec la technologie CMOS que nous retrouvons dans les appareils mobiles en raison de son faible coût et d'un encombrement réduit. Cette technologie permet d'acquérir rapidement l'image en exposant les lignes de l'image de manière séquentielle. Cependant cette méthode produit des déformations dans l'image s'il existe un mouvement entre la caméra et la scène filmée. Cet effet est connu sous le nom de «Rolling Shutter» et de nombreuses méthodes ont tenté de corriger ces artefacts.

Plutôt que de le corriger, des travaux antérieurs ont développé des méthodes pour extraire de l'information sur le mouvement à partir de cet effet. Ces méthodes reposent sur une extension de la modélisation géométrique classique des caméras pour prendre en compte l'acquisition séquentielle et le mouvement entre le capteur et la scène, considéré uniforme. À partir de cette modélisation, il est possible d'étendre le calcul de pose habituel (estimation de la position et de l'orientation de la scène par rapport au capteur) pour estimer aussi les paramètres du mouvement.

Dans la continuité de cette démarche, nous présenterons une généralisation à des mouvements non-uniformes basée sur un lissage des dérivées des paramètres de mouvement. Ensuite nous présenterons une modélisation polynomiale du «Rolling Shutter» et une méthode d'optimisation globale pour l'estimation de ces paramètres. Correctement implémenté, cela permet de réaliser une mise en correspondance automatique entre le modèle tridimensionnel et l'image. Pour terminer nous comparerons ces différentes méthodes tant sur des données simulées que sur des données réelles et conclurons.

Mots-clefs : caméras, «Rolling Shutter», modélisation géométrique, calcul de pose, estimation de mouvement, pose dynamique.

ABSTRACT

Title : Dynamic pose estimation with CMOS cameras using sequential acquisition

Computer Vision, a field of Computer Science, is about extracting information from cameras. Their sensors can be produced using the CMOS technology which is widely used on mobile devices due to its low cost and volume. This technology allows a fast acquisition of an image by sequentially exposing the scan-line. However this method produces some deformation in the image if there is a motion between the camera and the filmed scene. This effect is known as Rolling Shutter and various methods have tried to remove these artifacts.

Instead of correcting it, previous works have shown methods to extract information on the motion from this effect. These methods rely on a extension of the usual geometrical model of cameras by taking into account the sequential acquisition and the motion, supposed uniform, between the sensor and the scene. From this model, it's possible to extend the usual pose estimation (estimation of position and orientation of the camera in the scene) to also estimate the motion parameters.

Following on from this approach, we will present an extension to non-uniform motions based on a smoothing of the derivatives of the motion parameters. Afterwards, we will present a polynomial model of the Rolling Shutter and a global optimisation method to estimate the motion parameters. Well implemented, this enables to establish an automatic matching between the 3D model and the image. We will conclude with a comparison of all these methods using either simulated or real data.

Keywords : cameras, Rolling Shutter, geometrical model, pose estimation, motion estimation, dynamic pose.

REMERCIEMENTS

En premier lieu, je tiens à remercier Adrien pour son encadrement scientifique en tant que directeur de thèse, et pour la confiance qu'il m'a accordée depuis mes premiers pas dans l'équipe ComSee du LASMEA lors de mon stage de maîtrise jusqu'à la rédaction de ce mémoire. Je remercie également mon second encadrant, Omar, pour sa disponibilité, son support, ses idées et son aide technique. Ils ont su me laisser l'autonomie dont j'avais besoin, tout en étant présent et encourageant dans les moments de doute.

Cette recherche a été financée d'abord par le projet ANR Virago au sein du LASMEA, désormais Institut Pascal, que je remercie pour son accueil. Je remercie également l'Université d'Auvergne pour m'avoir permis de continuer ce travail dans le cadre d'un contrat d'ATER.

Je remercie également David Fofi et Thomas Corpetti d'avoir accepté d'être rapporteurs de ce mémoire, ainsi que Jean-Marc Lavest et Nicolas Andreff d'avoir donné leur accord pour participer au jury de soutenance. Je salue également Daniel Pizarro pour sa présence et son soutien.

Enfin je remercie mon entourage et toutes les personnes qui m'ont supporté et encouragé dans mon travail, que ce soit mes parents, des amis (notamment pour la relecture), des collègues (avec une mention spéciale pour Daniel et Baptiste qui m'ont bien aidé parfois).

TABLE DES MATIÈRES

TABLE DES FIGURES

CHAPITRE I

INTRODUCTION

I.1 Généralités : caméras et «Rolling Shutter»

Il existe de nos jours de nombreux systèmes d'imagerie, c'est à dire de systèmes qui servent à représenter une partie du monde réel sur une image. C'est le cas des caméras et appareils photographiques que nous connaissons tous, mais aussi de systèmes plus spécifiques comme les systèmes d'imagerie par résonance magnétique ou la radiographie aux rayons X. L'image issue d'un tel système, et en particulier des caméras, telle que nous la concevons en informatique est le résultat d'un processus parfois complexe mêlant entre autre système optique et circuit électronique (Cruset, 1966; Hedgecoe, 2009). Durant ce processus, de l'information sur la scène présente devant la caméra est enregistrée et peut être par la suite exploitée informatiquement. La branche de l'informatique qui s'intéresse aux problèmes de l'extraction de l'information à partir de capteurs d'imagerie est connue comme la vision par ordinateur («Computer Vision»). Cette information peut cependant être déformée ou détériorée par les différents composants du système, les effets résultant sont facilement observables sur l'image finale. Ainsi, qui n'a jamais pris une photographie floue ou contenant des déformations comme présentées dans la figure I.1 ? De telles déformations sont fréquentes avec les caméras des appareils numériques à faible coût, tels que ceux qui se trouvent dans les téléphones portables, tablettes et autres périphériques mobiles. Les différents éléments constituant une caméra et les dégradations sur l'image qui peuvent en résulter sont abordées au chapitre II.

L'étude des différents composants d'une caméra permet d'expliquer la majorité des artefacts présents habituellement dans une image informatique. Nous nous intéresserons plus particulièrement au circuit électronique qui reçoit la lumière et la convertit en information numérique, le capteur, et détaillerons son fonctionnement. Celui-ci peut en effet être à l'origine d'un phénomène causant des dé-

FIGURE I.1 – Exemple de déformations et détériorations de l'information contenue dans une image obtenue avec un appareil photographique de faible coût.

formations dans l'image qui sont très caractéristiques : scène penchée, étirée ou compressée, … Ce phénomène est nommé «Rolling Shutter» et il est provoqué par la technologie employée pour réaliser le capteur : la technologie CMOS («Complementary Metal Oxide Semiconductor»). Cette technologie est très prisée pour les appareils à faible coût ou les systèmes mobiles en raison des nombreux avantages qu'elle présente : très faible consommation électrique, simplicité du circuit et donc encombrement réduit, et un coût très avantageux. Pour la fabrication de capteur de caméra, elle est souvent opposée à la technologie CCD («Charge-Coupled Device») qui est généralement utilisée pour fabriquer les capteurs haut de gamme ou nécessitant un rapport signal sur bruit très élevé, ce dernier étant assez faible avec la technologie CMOS bien que les progrès récents sur cette technologie l'ait grandement amélioré. Une présentation de ces deux technologies et de leur mode de fonctionnement sera réalisée dans le chapitre III.

Les déformations provoquées par le «Rolling Shutter» sont aisément expliquées lorsque l'on connaît le fonctionnement du circuit qui récolte les photons et les convertit en signal électronique. En effet, dans ce type de capteur, les différentes lignes qui constituent la surface de collecte des photons peuvent être acquises de manière séquentielle. Lorsque la scène devant l'objectif est figée (et illuminée de manière constante) et que la caméra est fixe, l'image obtenue ne contient que les déformations ou détériorations résultant du bruit du capteur et du système optique que constitue l'objectif, comme c'est le cas avec un capteur «Global Shutter» qui expose toutes les lignes simultanément. En revanche si la caméra se déplace, que des objets de la scène sont mobiles ou que l'illumination change soudainement, alors des déformations ou artefacts apparaissent dans l'image. Un exemple typique de déformation résultant de

(a) Ici les piétons au bord de la route apparaissent penchés dans un sens, alors que le véhicule en mouvement est penché dans l'autre sens. Image de Jacques-Henri Lartigue.

(b) Exemple de correction possible de l'image permettant de retrouver un aspect normal des piétons et du véhicule. Image de Bert Otten aka. Lindolfi (www.imagingdna.com).

FIGURE I.2 – Exemple d'image obtenue avec un capteur de technologie CMOS utilisant le mode d'acquisition séquentielle lorsque la caméra est mobile et qu'un objet se déplace dans la scène. Cette image ayant été réalisée en 1912, il est évident qu'en réalité elle n'a pas été prise avec un appareil photographique employant la technologie CMOS, celle-ci n'ayant été exploitable industriellement qu'à partir des années 1960. Elle illustre néanmoins remarquablement bien le phénomène du «Rolling Shutter» puisqu'elle a été prise avec un appareil ICA qui provoque exactement le même effet bien que la raison en soit mécanique et non électronique. Un exemple de correction possible de l'effet du «Rolling Shutter» dans cette image fait apparaître une modification dans l'information contenue, ainsi la famille présente à l'arrière du véhicule dans l'image originale a presque disparu de l'image finale lors de la correction.

cette acquisition séquentielle est présenté dans la figure I.2. Dans cette thèse nous présenterons au chapitre III de manière détaillée les différentes déformations que peuvent causer le «Rolling Shutter». Ainsi que le montre la littérature, la problématique de la correction de ces déformations a été rapidement abordée lorsque la technologie CMOS a été massivement adoptée au début des années 2000, du fait de la nécessité de miniaturiser les appareils photographiques pour pouvoir les embarquer dans des systèmes mobiles. La figure I.2 présente également un exemple de telles corrections. Ces méthodes de correction rendent certes l'image esthétiquement plus correcte, mais elle conduit également bien souvent à une perte d'information comme toute méthode de correction. Nous présenterons succinctement un état de l'art des méthodes de corrections des effets du «Rolling Shutter» au chapitre III.

I.2 Modélisation géométrique et calcul de pose

La modélisation géométrique du phénomène «Rolling Shutter» nécessite d'abord de maîtriser la modélisation classique d'une caméra. De nombreuses hypothèses simplificatrices sont réalisées lors de cette modélisation (Hartley and Zisserman, 2003; Paragios et al., 2006; Forsyth and Ponce, 2003).

Ainsi, bien que l'objectif d'une caméra soit un assemblage complexe de lentilles, il est représenté par une simple lentille dont les caractéristiques sont celles équivalentes à l'ensemble du montage optique. De même, nous négligeons le caractère ondulatoire de la lumière et nous ne conservons donc que les lois de l'optique linéaire. Enfin l'hypothèse dite du trou d'épingle («pinhole») consiste à supposer que tous les rayons lumineux issus d'un point de la scène convergent en un seul point sur le capteur et donc il est possible de se limiter à ne considérer qu'un seul rayon lumineux par point de la scène, celui passant par le centre optique de l'objectif. En ignorant les distortions radiales dues à l'objectif, lesquelles sont facilement corrigées, le modèle géométrique ainsi obtenu est mathématiquement simple puisque la projection est dès lors modélisable, à un facteur d'échelle près, par des opérations de base de l'algèbre linéaire comme la multiplication matricielle (Baer, 1952; Coxeter, 2003). Cette modélisation sera développée en détail au chapitre II.

Une fois ce modèle défini, il reste à en trouver les paramètres propres à chaque système réel. Pour une caméra donnée, nous parlons d'étalonnage lorsque nous estimons ces paramètres (Tsai, 1986; Triggs, 1998; Zhang, 1999; Sturm and Maybank, 1999). Cela se réalise habituellement par l'observation d'objets, des mires d'étalonnage, dont nous connaissons la structure tridimensionnelle et qui facilite la mise en correspondances avec la projection dans l'image. De nombreuses boites à outils existent pour réaliser cette tâche (Bouguet, 2010). Les calculs nécessaires pour cela reposent sur des outils algébriques présentés dans les annexes B et C ainsi que sur des méthodes d'optimisation développées dans l'annexe A. Parmi les paramètres estimés lors de l'étalonnage se trouvent la position et l'orientation entre la caméra et l'objet utilisé. Une fois les autres paramètres connus, nous pouvons nous intéresser à la problématique de retrouver cette position et orientation pour un objet quelconque dont nous connaissons la structure tridimensionnelle. Cela est connu sous le nom de calcul de pose. De nombreuses méthodes plus ou moins récentes ont déjà été présentées pour le cas du «Global Shutter» (Dementhon and Davis, 1995; Ameller et al., 2002; Lepetit et al., 2009), nous en présenterons quelques unes au cours du chapitre II.

I.3 Travaux présentés

Avec une caméra disposant du «Rolling Shutter» et configurée pour réaliser une acquisition séquentielle des lignes de l'image, la problématique du calcul de pose devient plus complexe. En effet, la position et l'orientation de l'objet ne sont plus fixes dans le temps, mais varient selon l'endroit où l'on se place dans l'image. Nous proposerons donc au chapitre IV une modélisation générique de ce phénomène que nous nommerons *pose dynamique*.

La problématique du calcul de pose dynamique n'est pas nouvelle (Meingast et al., 2005; Ait-Aider et al., 2006), bien qu'elle soit très récente, elle a notamment déjà été traitée par plusieurs auteurs de ma-

nière différentes pour un mouvement entre l'objet et la caméra qui soit uniforme. Nous réaliserons un état de l'art des méthodes déjà existantes lors du chapitre III. De nombreuses questions peuvent néanmoins encore se poser à ce sujet. Nous apporterons par exemple au chapitre IV une nouvelle proposition pour modéliser la projection «Rolling Shutter» lorsque le mouvement n'est pas uniforme, en y associant une méthode de calcul de pose dynamique reposant sur une optimisation non-linéaire présentée dans l'annexe A.

Les travaux précédents sur le calcul de pose dynamique reposaient également sur l'optimisation non-linéaire, dont un des inconvénients est de ne pouvoir réaliser qu'une optimisation locale, ce qui suppose de pouvoir calculer une initialisation de la pose dynamique qui soit proche du minimum. Les méthodes précédentes réalisaient cette initialisation de manière empirique à partir d'un calcul de pose «Global Shutter». Nous proposerons ainsi dans le chapitre IV une nouvelle méthode de calcul linéaire pour cette initialisation dans le cas non-uniforme.

Une autre alternative à la recherche d'une initialisation est de réaliser une optimisation globale, ce qui est rendu possible en modélisant la projection «Rolling Shutter», dans le cas d'un mouvement uniforme, de manière polynomiale. Ceci est réalisable sous certaines conditions sur le mouvement qui sont aisément satisfaites ou très proches de l'être dans la plupart des applications réelles. L'optimisation polynomiale qui en résulte peut être traitée de manière efficace et globale par des méthodes récentes (Kukelova et al., 2008; Henrion et al., 2009), comme celles présentées dans l'annexe A, notamment en utilisant les capacités de calcul parallèles des ordinateurs modernes. Le chapitre V couvrira cette modélisation et précisera comment implémenter efficacement le calcul de pose dynamique.

L'assurance d'obtenir le minimum global apportée par l'optimisation polynomiale, et le temps réduit nécessaire pour cela grâce à la parallélisation, permet d'envisager la mise en correspondance automatique des points d'intérêt de l'image avec les points du modèle tridimensionnel de l'objet. Le modèle tridimensionnel de l'objet est obtenu par une paire de caméras stéréoscopique, comme montré en annexe D. Après la reconstruction stéréoscopique, les correspondances entre les points de l'objet et leur projection dans les images utilisées pour la reconstruction sont connues et peuvent servir de patron pour mettre en correspondance les points tridimensionnel et leur projection dans une nouvelle image. Cette mise en correspondance automatique pouvant générer des correspondances erronées, il est alors nécessaire de rendre robuste l'estimation de la pose dynamique, ce qui peut être réalisé à l'aide de méthodes comme le «RANdom SAmple Consensus» (Fischler and Bolles, 1981). Tout cela sera détaillé au chapitre V.

Dans le chapitre VI, ces différents travaux seront comparés entre eux et face aux travaux précédents sur un ensemble de données simulées, ce qui a nécessité de s'intéresser à la problématique de la création de ces données. Nous proposerons ainsi deux méthodes pour créer de telles données. Des essais sur des collections d'images réelles seront aussi présentés pour valider le principe de la mise en correspondance automatique. Nous conclurons au chapitre VII par un retour sur les travaux qui seront présentés dans

cette thèse et aborderons quelques perspectives de recherches futures.

CHAPITRE II

LES CAMÉRAS ET LEUR MODÉLISATION

Résumé

Les caméras numériques modernes sont des systèmes complexes mêlant optique et électronique afin de permettre la création d'un signal numérique à deux dimensions interprétable comme image (Hedgecoe, 2009). En considérant une dimension supplémentaire, le temps, nous obtenons alors la vidéo qui n'est rien d'autre qu'une succession d'images à deux dimensions. Un minimum de connaissances concernant le fonctionnement des caméras est nécessaire afin de correctement cerner la problématique qui occupera les chapitres suivants. Afin d'avoir un vocabulaire clair et commun par la suite, le début de ce chapitre est dédié à cet aspect. Nous nous intéresserons en deuxième partie à la modélisation mathématique des caméras, définissant ainsi la notion de matrice de projection d'une caméra qui contient l'information sur la position et l'orientation de la caméra ainsi que certains paramètres internes. La troisième partie est dédiée à l'étalonnage et au calcul de pose, qui consiste à calculer tout ou partie des informations contenues dans la matrice de projection. Enfin nous verrons en quatrième partie comment il est possible, avec les méthodes actuelles, de mettre en correspondance les projections d'un même point de la scène filmée sur deux images prises d'un point de vue différent.

II.1 La création de l'image

Dans le processus menant à la création de l'image, nous distinguons quatre composantes majeures sur lesquelles nous allons nous attarder et qui sont illustrées dans la figure II.1. Face à la caméra, nous trouvons tout d'abord la scène filmée qui est exposée à une source de lumière, naturelle ou artificielle. La lumière se réfléchit en partie sur la matière qui compose la scène. C'est cette lumière réfléchie qui, vue à travers l'objectif, est transformée en image optique [1]. Enfin le capteur crée le signal électronique correspondant à l'image optique. L'objectif et le capteur ont des propriétés propres qui influent sur l'image obtenue, nous les détaillerons.

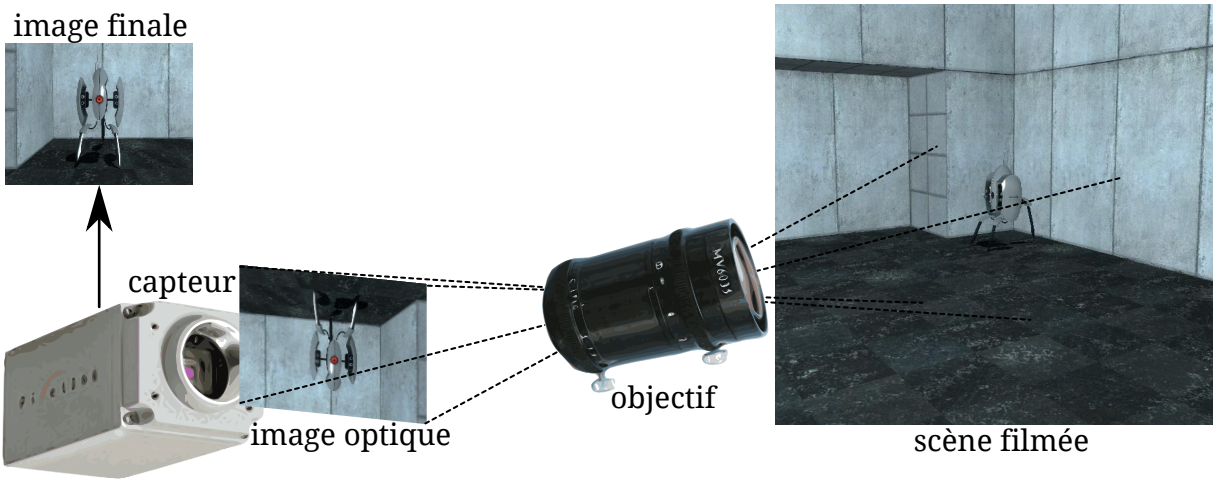


FIGURE II.1 – La scène filmée, qui se trouve dans le champ de la caméra, est transformée en une image optique par l'objectif. Le capteur de la caméra transforme cette image optique en un signal électronique puis numérique pour obtenir l'image finale.

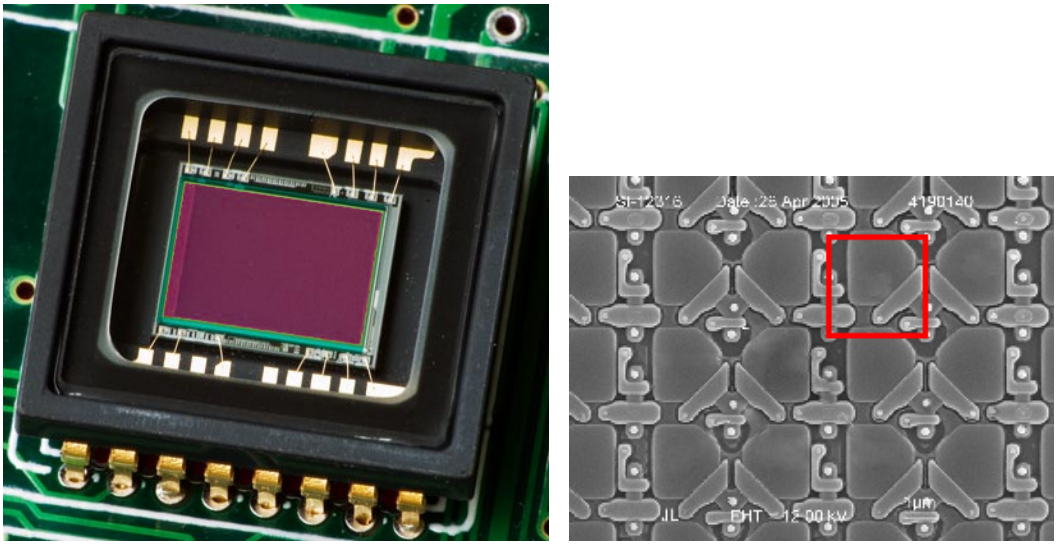
II.1.1 Le capteur et la rétine

Le capteur d'une caméra est constitué de trois parties (Photographie, 2013; Luxorion, 2013) :

- Une rétine qui est une surface contenant un ensemble de cellules photosensibles produisant une charge électrique lorsqu'elles sont exposées à la lumière ;
- Un circuit électronique collectant les charges électriques au sein de la rétine ;
- Un circuit d'interconnexion avec les circuits de contrôle de la caméra.

Un exemple de présentation d'un capteur se trouve dans la figure II.2. Sur la vue macroscopique II.2(a), nous retrouvons la rétine, qui est ici de couleur violette, ainsi que le circuit d'interconnexion, qui est

1. On fera la distinction entre l'image, qui est le résultat final délivré par la caméra, et l'image optique qui est le résultat du passage de la lumière dans le système optique constituant l'objectif.



(a) Vue macroscopique du capteur.

(b) Vue microscopique de la rétine.

FIGURE II.2 – Vue générale d'un capteur de caméra numérique et vue au microscope électronique de sa surface, la rétine. Nous y distinguons les cellules photosensibles, à l'intérieur du carré rouge, et le circuit de collecte, le long des bords rouges. La taille du carré rouge est ici d'environ $4\mu m$.

ici la fine partie verte entourant la rétine. Les petits fils dorés réalisent la jonction entre ce circuit et les circuits de contrôle de la caméra. La rétine vue au microscope électronique peut être observée sur la figure II.2(b). Nous y retrouvons les cellules photosensibles, qui sont les grands demi-disques, et le circuit de collecte des charges, qui est formé des petits éléments entre les cellules. Le fonctionnement de ce circuit sera étudié en détail dans le chapitre III puisqu'il est à l'origine de la problématique étudiée ici.

Les deux premières propriétés importantes d'un capteur sont la taille de la rétine, exprimée en millimètre, et sa définition, c'est-à-dire le nombre de cellules photosensibles qu'elle contient. La résolution de la rétine est le rapport entre le nombre de cellules et sa surface et s'exprime en «dot per inch» (DPI) qui donne le nombre de cellules dans un carré d'un pouce de large. Plus la résolution est grande et plus les cellules photosensibles sont petites, et donc plus les détails de l'image sont fins.

Une autre propriété importante est la sensibilité des cellules photosensibles à la quantité de lumière reçue. Plus cette sensibilité est importante, c'est-à-dire qu'un petit changement de la quantité de lumière reçue implique un changement de la charge électrique, plus le capteur captera les nuances de la scène observée. Ceci n'est vrai que si le pas de quantification, représentant la plus petite valeur de tension discernable lors de la conversion en signal numérique, est suffisamment faible.

Le capteur peut produire des détériorations de l'image, par exemple lorsque la quantité de lumière atteignant une des cellules est trop importante, celle-ci se retrouve saturée. Dans ce cas et selon la

FIGURE II.3 – Exemple d'éblouissement du capteur, les cellules photosensibles exposées au centre de la source lumineuse sont saturées et leurs charges électriques excédentaires sont transférées aux cellules voisines. Cela peut se propager sur de longues distances, créant ainsi les six raies lumineuses visibles ici.

technologie utilisée dans le capteur, il peut arriver que l'excès de charge électrique déborde sur les cellules voisines créant un effet d'éblouissement. Un exemple est donné dans la figure II.3. Ce type de défaut est de plus en plus souvent corrigé dans les capteurs moderne par la présence d'un circuit drainant les charges en excès.

II.1.2 L'objectif

L'objectif est un dispositif optique convergent constitué de plusieurs lentilles permettant de former une image optique de ce qui se trouve devant lui (Forsyth and Ponce, 2003; Cruset, 1966). Afin de pouvoir modéliser simplement les caméras, nous considérerons que ce système optique respecte les conditions de Gauss [2] (Pérez, 2004) et que sa réponse est donc linéaire. Nous pouvons ainsi appliquer les lois de l'optique géométrique (Moussa and Ponsonnet, 1975). L'image optique d'un point de la scène se situe à l'endroit où les différents rayons lumineux issus de ce point convergent à travers l'objectif. Pour les points qui se trouvent vraiment très éloignés de l'objectif, qui sont alors dits «à l'infini», leurs images optiques se trouvent au foyer de l'objectif. Pour tout point de la scène, il existe un rayon lumineux qui traverse l'objectif sans être dévié par celui-ci. L'ensemble des rayons lumineux qui traversent l'objectif sans être déviés se croisent en un point qui est le centre optique de l'objectif. La distance entre le foyer et le centre optique de l'objectif est nommée distance focale. La figure II.4 présente ces différentes notions.

La rétine du capteur se trouve juste après l'objectif et une propriété importante de l'ensemble est la distance qui se trouve entre le centre optique de l'objectif et la rétine du capteur (Kieffer, 1985). L'apparence des images optiques des points de la scène au niveau de la rétine est fonction de celle-ci : il s'agit de taches plus ou moins étendues, comme il est illustré dans la figure II.5. Lorsqu'elle est ajustée pour que la tâche d'un point donné de la scène se réduise au minimum, nous parlons alors de mise au point.

L'objectif n'est pas constitué que de lentilles, il contient également un élément mécanique nommé

2. Pour satisfaire les conditions de Gauss, il faut que les rayons lumineux qui traversent le système optique gardent des angles d'incidence faibles et restent proches de l'axe optique.

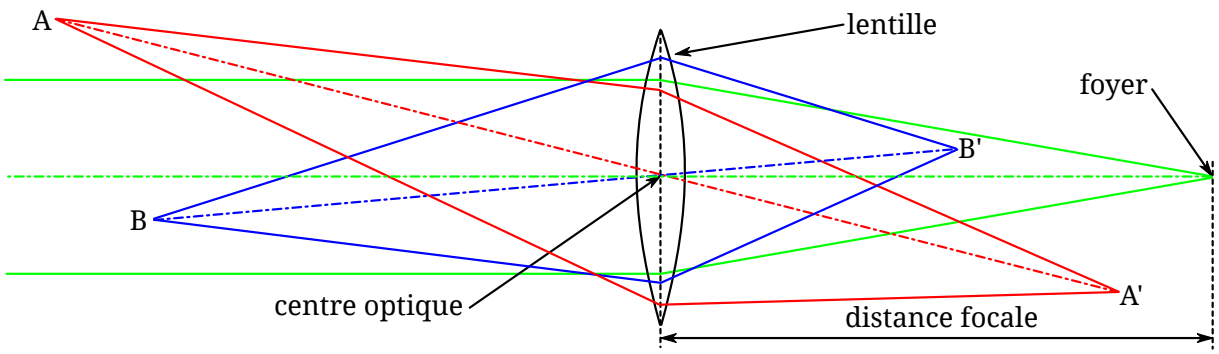


FIGURE II.4 – Si nous simplifions l'objectif à une simple lentille, ce schéma présente les différents éléments importants et le principe de son fonctionnement. Les rayons lumineux issus du point A convergent sur l'image optique située en A', tandis que ceux issus du point B convergent en B'. En vert sont représentés les rayons lumineux issus d'un point à l'infini et qui convergent donc vers le foyer de l'objectif définissant ainsi la distance focale jusqu'au centre optique où s'intersectent les rayons lumineux des différents points.

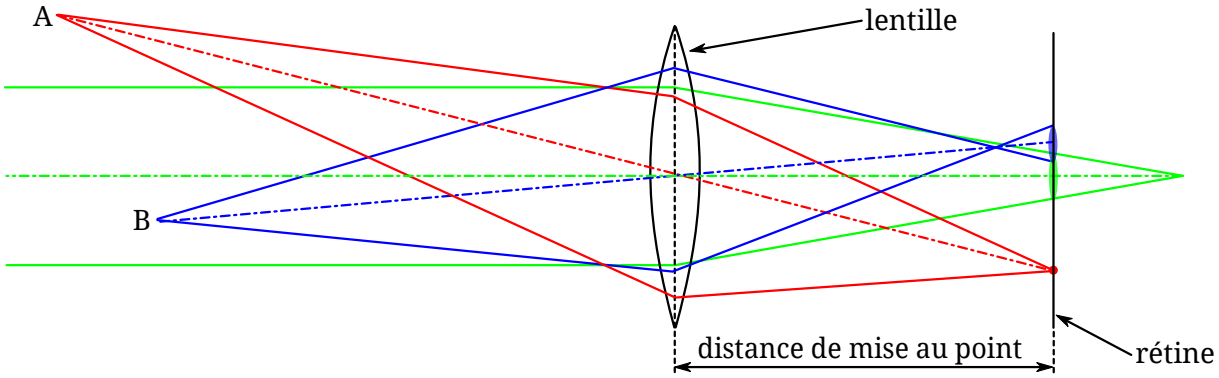


FIGURE II.5 – Lorsque nous plaçons la rétine derrière l'objectif, les images des points de la scène sur celle-ci sont des taches plus ou moins larges. La mise au point consiste à placer la rétine à une distance telle que certains des points de la scène correspondent à des taches les plus fines possibles et qu'ils semblent ainsi nets sur l'image finale.

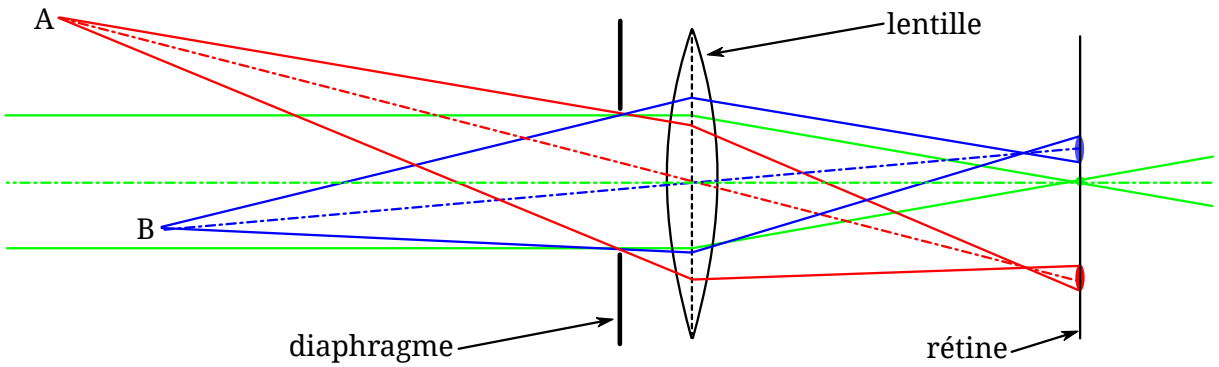


FIGURE II.6 – La fermeture du diaphragme permet de limiter les rayons lumineux à ceux traversants la lentille proche du centre. Cela a pour effet de diminuer la taille des taches issues des points de la scène sur la rétine du capteur et donc d'augmenter la profondeur de champ. Elle réduit cependant la luminosité globale de l'image.

diaphragme et dont l'ouverture est variable. Le but premier de ce mécanisme est de restreindre les rayons lumineux à ceux passant proche du centre de la lentille, afin d'être au plus près des conditions de Gauss. La mise au point doit cependant se faire en tenant compte de l'ouverture du diaphragme. En effet la distance de mise au point est légèrement décalée lorsque l'ouverture du diaphragme change, comme illustré figure II.6. De plus en réduisant le nombre de rayons lumineux, la taille des taches issues des points de la scène diminue également. Ceci a pour effet d'augmenter l'espace de la scène dont l'image est nette au niveau de la rétine, nous parlons alors de profondeur de champ. Enfin en limitant le nombre de rayons lumineux provenant des différents points de la scène, le diaphragme a également un effet sur la luminosité de l'image : celle-ci augmente avec son ouverture.

L'objectif peut introduire des distorsions géométriques dans l'image (Forsyth and Ponce, 2003). Elles peuvent être provoquées par des défauts d'alignement des lentilles, nous parlons alors de distorsion tangentielle, mais aussi tout simplement par la forme symétrique et sphérique des lentilles, c'est la distorsion radiale. La distorsion tangentielle est généralement faible sur les objectifs classiques et récents. En revanche la distorsion radiale est souvent présente sur certains objectifs à courte focale destinés à être utilisés pour avoir un grand angle de champ. Elle est même parfois volontairement présente à des fins artistiques comme sur les objectifs dits «fisheye». Deux exemples de distorsion sont donnés dans la figure II.7

Deux autres conséquences de la présence de l'objectif peuvent encore être observées. Tout d'abord les aberrations chromatiques qui sont le résultat de la décomposition du spectre lumineux par l'objectif à la manière d'un prisme, ou encore des traitements appliqués au verre composant les lentilles comme les traitements anti-reflets. Elles se retrouvent surtout sur les images couleurs. Enfin lorsque l'ouverture est extrêmement faible, il peut se produire de la diffraction au niveau du diaphragme et alors le modèle de l'optique linéaire utilisé habituellement pour modéliser la projection des caméras n'est plus valide.

(a) Objectif grand angle. Image d'Aurélie Fruitière.

(b) Objectif «fisheye». Image de www.boreally.org.

FIGURE II.7 – Exemples de distorsions radiales dues à des objectifs particuliers. Ces distorsions ont pour conséquence de déformer les droites de la scène pour donner des courbes. Plus nous nous éloignons du centre de distorsion et plus les déformations sont prononcées.

II.1.3 La scène filmée

La taille de la rétine du capteur utilisé et la distance focale de l'objectif permettent de définir un angle nommé «angle de champ» qui délimite autour de l'axe optique une zone de l'espace appelée champ de la caméra. La relation reliant ces différentes propriétés est donnée par (Forsyth and Ponce, 2003; Photographie, 2013) :

$$\tan\left(\frac{\theta}{2}\right) = \frac{x}{2f}, \tag{II.1}$$

où θ est l'angle de champ en largeur, hauteur ou diagonale selon que x soit la largeur, la hauteur ou la diagonale de la rétine. f désigne la distance focale de l'objectif. La scène filmée est alors constituée de tout ce qui se trouve dans le champ de la caméra.

Elle peut être constituée uniquement d'objets immobiles, auquel cas la scène est dite statique ou rigide. Filmer un bâtiment ou un paysage est un exemple de scène statique. Si des objets de la scène filmée sont amenés à se déplacer ou à se déformer, nous parlons alors de scène dynamique ou déformable. Une voiture passant devant une caméra fournit un exemple de scène dynamique.

Selon l'éclairage, la scène peut également être lumineuse ou sombre. Il est particulièrement important d'en tenir compte pour régler l'ouverture de l'objectif et le temps d'exposition de la rétine du capteur de manière adéquate afin d'obtenir une image correctement exposée.

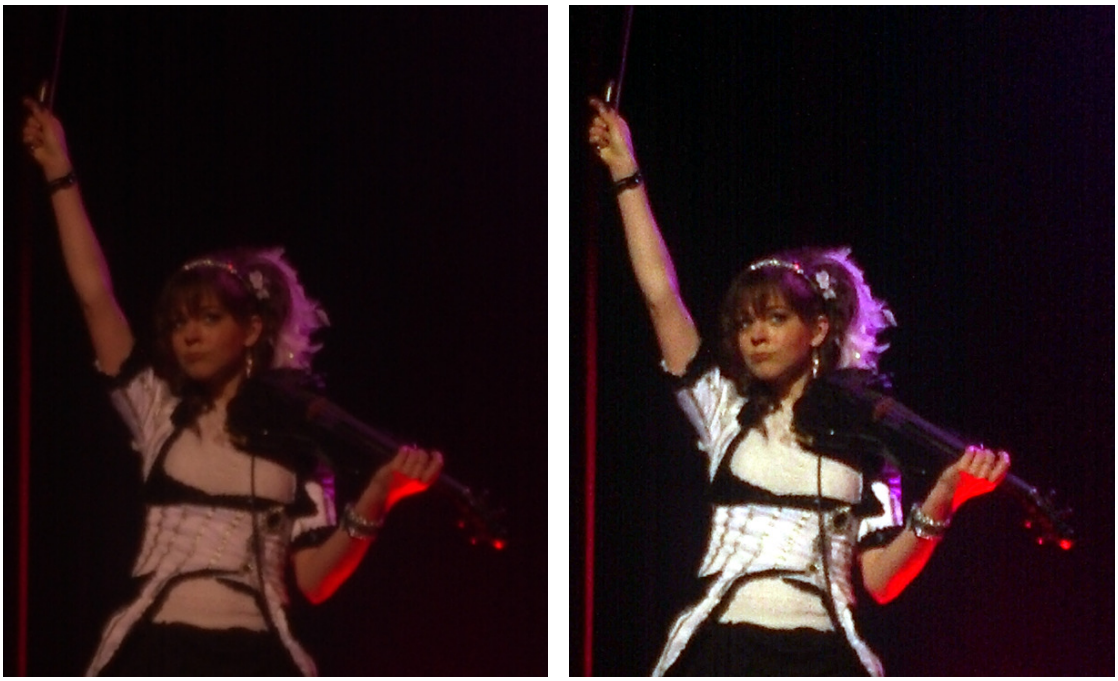
II.1.4 L'image

L'image finale au format numérique est constituée d'une suite d'informations binaires interprétable comme une suite de pixels comportant chacun la luminosité à laquelle a été exposé la cellule photosensible correspondante (Luxorion, 2013). Les pixels sont généralement repérés dans l'image par la ligne et la colonne où ils se trouvent. Le repère dans lequel est exprimé la position des pixels est donc centré sur le coin supérieur gauche de l'image, le premier axe étant orienté vers le bas et le deuxième vers la droite.

En fonction des réglages de la caméra et de la scène filmée, le rendu de l'image peut être de plus ou moins bonne qualité. Divers algorithmes de traitement d'image peuvent être appliqués pour améliorer le résultat. Ces améliorations se font souvent vis-à-vis d'un critère donné, et détériorent souvent d'autres aspects. Le traitement d'image n'étant pas le thème de cette thèse, nous supposerons que les images que nous manipulons ont un rendu correct, qu'elles soient brutes ou retouchées. Citons à titre d'exemples quelques méthodes que nous pouvons utiliser dans ce but :

- Les méthodes de réhaussement de contour dont l'objectif est de rendre une image légèrement floue un peu plus nette. Elles tendent également à réhausser le bruit de l'image de fait.
- Les différents filtrages (moyenneur, gaussien, median, …) qui servent à réduire le bruit d'une image, bien souvent au prix de la netteté.
- Les traitements de l'histogramme des couleurs (correction gamma, égalisation, …) qui permettent bien souvent de corriger une image mal contrastée à cause d'un temps d'exposition inadapté.

Un exemple de retouches est donné dans la figure II.8.



(a) Image originale (b) Image retouchée

FIGURE II.8 – L'image originale est peu contrastée, légèrement floue, et la balance des blancs est incorrecte. Un traitement de l'histogramme et un réhaussement de contour permet d'améliorer l'image.

II.2 Modélisation géométrique de la projection

II.2.1 Hypothèse «trou d'épingle»

La modélisation de la projection d'un point à travers l'objectif sur la rétine du capteur d'une caméra serait trop complexe si elle devait être totalement fidèle à la réalité. Comme mentionné précédemment, nous considèrons que l'objectif répond aux lois de l'optique linéaire et peut donc être modélisé géométriquement. Afin de conserver un modèle mathématique de la projection relativement simple, une deuxième approximation est très souvent réalisée. Celle-ci consiste à ne considérer qu'un seul rayon lumineux par point de la scène projeté, celui passant par le centre optique de l'objectif (Cruset, 1966; Forsyth and Ponce, 2003).

Cette hypothèse, qui est illustrée dans la figure II.9, se justifie tout-à-fait lorsque la profondeur de champ est grande et que les points de la scène se projettent donc de manière nette sur la rétine. Cela est donc particulièrement vrai lorsque l'ouverture du diaphragme est relativement faible. Dans cette situation le diaphragme joue le même rôle qu'un minuscule trou d'épingle («pinhole») juste assez grand pour ne laisser ainsi passer qu'un seul rayon lumineux par point de la scène, mais encore assez grand pour ne pas causer de diffraction.

II.2.2 Notations mathématiques

Le corps des réels muni de ses opérations élémentaires habituelles est noté classiquement $\mathbb{R}$. Les éléments de $\mathbb{R}$ sont notés en italique, comme par exemple c. Le sous-ensemble des nombres naturels est noté $\mathbb{N}$. Nous notons tout autre ensemble par une police calligraphique comme par exemple $\mathcal{E}$. Lorsqu'un ensemble $\mathcal{E}$ est privé de son élément nul, nous le notons $\mathcal{E}^*$.

Les matrices à coefficients dans $\mathbb{R}$ sont notées avec une police sans-serif, comme M par exemple. La transposée d'une matrice M est notée $\mathsf{M}^\top$ et sa trace $\mathrm{tr}(\mathsf{M})$. La matrice identité est notée $\mathtt{I}$, et la matrice nulle $\mathsf{0}$. Les vecteurs d'un espace vectoriel sur $\mathbb{R}$ sont notés en gras, comme $\mathbf{v}$. Ils sont représentés par des matrices colonnes, par exemple dans $\mathbb{R}^3$

$$\mathbf{v} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

représente un vecteur. Le vecteur nul est noté $\mathbf{0}$. Le produit scalaire de deux vecteurs $\mathbf{a}$ et $\mathbf{b}$ est donc naturellement noté $\mathbf{a}^\top\mathbf{b}$. La norme euclidienne d'un vecteur $\mathbf{v}$, aussi appelée norme deux, est notée $\|\mathbf{v}\|$

Le groupe des rotations tridimensionnelles est noté $\mathcal{SO}(3)$. Il est formé des matrices telles que $\mathsf{M}^\top\mathsf{M} = \mathtt{I}_{3\times3}$ et $\det(\mathsf{M}) = 1$ où det est la fonction qui associe son déterminant à une matrice. La matrice antisymétrique associée au vecteur $\mathbf{v} = \begin{bmatrix} x & y & z \end{bmatrix}^\top \in \mathbb{R}^3$ est notée

$$[\mathbf{v}]_\wedge = \begin{bmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{bmatrix}.$$

II.2.3 Coordonnées homogènes

Les coordonnées homogènes (Baer, 1952; Coxeter, 2003) sont une construction mathématique qui permet de modéliser aisément la projection centrale résultant de l'hypothèse «trou d'épingle» (Hartley and Zisserman, 2003; Forsyth and Ponce, 2003). Dans un espace vectoriel $\mathcal{E}$ de dimension n donné et privé de son élément nul, des classes d'équivalence par homothétie des vecteurs non nuls sont créées. Deux vecteurs $\mathbf{v}$ et $\mathbf{w}$ non nuls de $\mathcal{E}$ sont dits équivalents si il existe un scalaire c non nul tel que $\mathbf{v} = c\mathbf{w}$. Nous écrivons alors $\mathbf{v} \sim \mathbf{w}$ l'équivalence de ces deux vecteurs. Il faut remarquer que tous les vecteurs équivalents à un vecteur donné ont pour support la même droite vectorielle passant par l'origine. Ceci permet donc de représenter l'ensemble des droites vectorielles passant par l'origine de l'espace vectoriel par le choix d'un des vecteurs de la classe d'équivalence.

Si nous considérons les coordonnées dans une base canonique de l'espace vectoriel $\mathcal{E}$ de n'importe quel vecteur de la classe d'équivalence, nous obtenons des coordonnées dites homogènes qui représentent la direction de la droite vectorielle. Pour différencier un vecteur classique $\mathbf{v}$ de l'espace vectoriel d'un vecteur qui correspond à des coordonnées homogènes, nous notons ce dernier $\tilde{\mathbf{w}}$. Puisque les différents vecteurs de la classe sont équivalents, il est possible de choisir n'importe lequel pour représenter cette direction, mais nous privilégions habituellement le vecteur dont la dernière coordonnée est 1. Cela revient à fixer un hyperplan [3] $\mathcal{F}$ ne passant pas par l'origine dans lequel il est possible d'identifier une droite de l'espace vectoriel par un unique jeu de coordonnées. Dans le cas de l'espace vectoriel $\mathbb{R}^3$, cet hyperplan est un plan qui s'appelle plan projectif.

Pour tout vecteur de coordonnées homogènes $\tilde{\mathbf{p}} = \begin{bmatrix} p_1 & \dots & p_{a-1} & p_a \end{bmatrix}^\top$ de l'espace vectoriel $\mathcal{E}$ tel que $p_a \neq 0$, nous obtenons les coordonnées dans $\mathcal{F}$ représentant la droite vectorielle correspondante en utilisant la fonction $\boldsymbol{\varphi}$ définie par

$$\boldsymbol{\varphi}(\tilde{\mathbf{p}}) = \begin{bmatrix} p_1/p_a & \dots & p_{a-1}/p_a \end{bmatrix}^\top, \tag{II.2}$$

La figure II.10 illustre ce concept pour l'espace vectoriel $\mathbb{R}^3$. À partir des coordonnées $\mathbf{p}$ dans $\mathcal{F}$ d'une

3. Dans notre cas et pour simplifier, un hyperplan est un sous-espace vectoriel de dimension $n-1$.

droite vectorielle, il est possible de retrouver un des vecteurs $\tilde{\mathbf{p}}$ de la classe d'équivalence correspondante en ajoutant à ces coordonnées dans $\mathcal{F}$ une coordonnée supplémentaire qui est unitaire. Par exemple dans l'espace vectoriel $\mathbb{R}^3$, au point $\mathbf{v} = \begin{bmatrix} x & y \end{bmatrix}^\top$ du plan projectif, nous associons les coordonnées homogènes notées $\tilde{\mathbf{v}} = \begin{bmatrix} x & y & 1 \end{bmatrix}^\top$.

De par la définition II.2 de la fonction φ, nous pouvons remarquer que si la dernière coordonnée homogène était nulle, tout vecteur associé à la droite vectorielle dans $\mathcal{E}$ est infini. Les coordonnées homogènes de dernière coordonnée nulle représentent donc les droites vectorielles passant par un point à l'infini, les premières coordonnées indiquant la direction de cet infini.

II.2.4 La projection perspective

Pour modéliser la projection perspective, à l'instar de (Hartley and Zisserman, 2003; Paragios et al., 2006; Forsyth and Ponce, 2003), considérons un repère tridimensionnel $(\mathbf{C}, \vec{i}, \vec{j}, \vec{k})$ de la caméra dont l'origine $\mathbf{C}$ soit le centre optique de l'objectif, l'axe $\vec{k}$ soit l'axe orthogonal à la rétine du capteur passant par le centre optique et les axes $\vec{i}$ et $\vec{j}$ soient parallèles aux bords de la rétine. Notons f la distance orthogonale entre la rétine et le centre optique que nous appellerons désormais la distance focale [4]. La figure II.11 illustre le positionnement de ces éléments. Nous pouvons remarquer que tant que le plan de projection est situé à cette distance, il peut être considéré comme étant devant ou derrière le centre optique. La différence n'est que d'une symétrie centrale par l'origine des projections. Pour simplifier les illustrations, nous le placerons toujours devant le centre optique bien qu'en réalité il soit toujours derrière.

Soit $\mathbf{q}$ la projection sur la rétine d'un point $\mathbf{p}$ de la scène. Le point $\mathbf{q}$ étant sur la rétine il peut donc s'écrire sous la forme $\mathbf{q} = \begin{bmatrix} a & b & f \end{bmatrix}^\top$ dans le repère caméra. De par l'hypothèse «trou d'épingle», le centre optique, $\mathbf{q}$ et $\mathbf{p}$ sont sur une même droite vectorielle : $\mathbf{q} \sim \mathbf{p}$ puisque $\mathbf{C} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^\top$. En

4. Comme nous l'avons vu section II.1.2, la distance focale telle qu'elle est définie en optique est la distance entre le centre optique et le foyer image. Néanmoins dans le domaine de la vision par ordinateur, nous désignons habituellement par le terme «distance focale» celle entre la rétine et le centre optique, et que nous avons précédemment appelée distance de mise au point.

écrivant $\mathbf{p} = \begin{bmatrix} x & y & z \end{bmatrix}^\top$ nous avons alors

$$\exists \lambda \in \mathbb{R}^*, \begin{cases} a = \lambda x \\ b = \lambda y \\ f = \lambda z \end{cases} \tag{II.3}$$

$$\Leftrightarrow \lambda = \frac{a}{x} = \frac{b}{y} = \frac{f}{z} \tag{II.4}$$

$$\Leftrightarrow \begin{cases} a = f\frac{x}{z} \\ b = f\frac{y}{z} \end{cases} . \tag{II.5}$$

En utilisant les coordonnées homogènes, il est possible d'écrire cette projection sous forme matricielle. En effet

$$\begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{p} = \begin{bmatrix} fx \\ fy \\ z \end{bmatrix} = \begin{bmatrix} az \\ bz \\ z \end{bmatrix} \tag{II.6}$$

et alors

$$\boldsymbol{\varphi}\left(\begin{bmatrix} az \\ bz \\ z \end{bmatrix} \right) = \begin{bmatrix} a \\ b \end{bmatrix} \tag{II.7}$$

donne la position, que nous noterons $\mathbf{q}' = \begin{bmatrix} a & b \end{bmatrix}^\top$, de la projection de $\mathbf{p}$ sur la rétine exprimée dans un repère centré sur le point principal. Ce dernier est défini comme la projection orthogonale du centre optique sur la rétine.

Lorsque les coordonnées de $\mathbf{p}$ sont exprimées dans un repère différent du repère caméra, nommé habituellement repère monde ou objet, il faut alors ajouter un changement de repère avant la projection :

$$\tilde{\mathbf{q}}' = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathsf{R} & \mathbf{t} \end{bmatrix} \tilde{\mathbf{p}}, \tag{II.8}$$

où $\begin{bmatrix} \mathsf{R} & \mathbf{t} \end{bmatrix}$ représente la pose de la caméra dans le repère monde, c'est-à-dire la position et l'orientation du repère caméra dans le repère monde.

Comme il a été vu dans la section II.1.4, le repère de l'image est souvent centré sur le coin supérieur gauche et les pixels sont repérés par leur ligne et colonne. En conséquence il faut opérer un changement

de repère et d'échelle vers ce repère pour aboutir au modèle final suivant :

$$\tilde{\mathbf{m}} = \underbrace{\begin{bmatrix} k_u f & 0 & u_0 \\ 0 & k_v f & v_0 \\ 0 & 0 & 1 \end{bmatrix}}_{=\mathsf{K}} \begin{bmatrix} \mathsf{R} & \mathbf{t} \end{bmatrix} \tilde{\mathbf{p}}, \tag{II.9}$$

où k_u et k_v sont respectivement les résolutions horizontale et verticale exprimées en pixels par millimètre. Généralement il est posé $\alpha_u = k_u f$ et $\alpha_v = k_v f$ qui sont les résolutions horizontale et verticale exprimées en pixels. Nous appelons habituellement ces deux valeurs les focales effectives. Le vecteur $\begin{bmatrix} u_0 & v_0 \end{bmatrix}^\top = \mathbf{c}_0$ est la position du point principal dans le repère image. La matrice K est appelée matrice d'étalonnage et reflète les paramètres internes de la caméra. Le produit de la matrice d'étalonnage par la matrice de pose donne la matrice de projection finale : $\mathsf{M} = \mathsf{K} \begin{bmatrix} \mathsf{R} & \mathbf{t} \end{bmatrix}$. Une illustration de ce modèle au complet est donnée par la figure II.12.

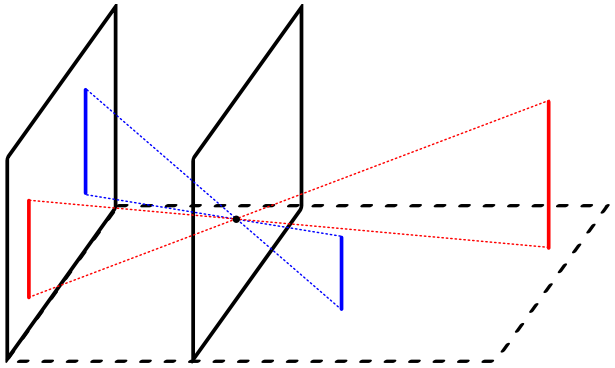


FIGURE II.9 – L'objectif est considéré comme un trou d'épingle par lequel ne peut passer qu'un seul rayon lumineux pour chaque point de la scène. Il est possible de noter que l'effet de la perspective donne l'impression sur la rétine du capteur que l'objet bleu et l'objet rouge sont presque de même taille.

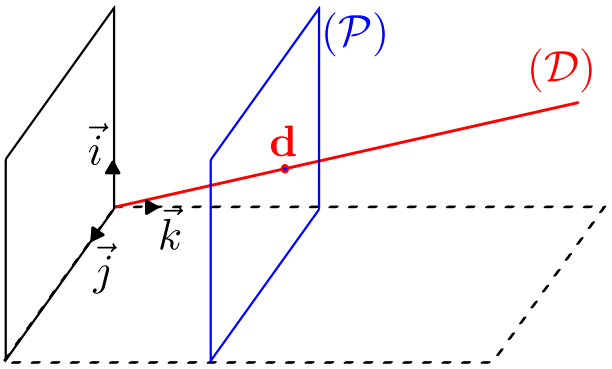


FIGURE II.10 – Dans l'espace vectoriel $\mathbb{R}^3$, tous les vecteurs non nuls ayant pour support la droite vectorielle $(\mathcal{D})$ sont équivalents. La projection sur le plan projectif $(\mathcal{P})$ ne passant pas par l'origine est représentée par le point $\mathbf{d}$ du plan $(\mathcal{P})$ qui intersecte la droite $(\mathcal{D})$.

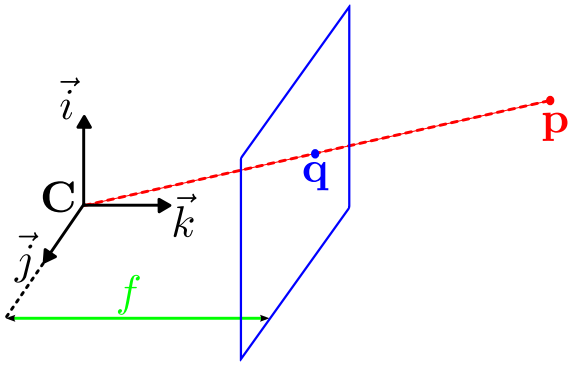


FIGURE II.11 – Le repère caméra $(\mathbf{C}, \vec{i}, \vec{j}, \vec{k})$ et le plan de projection sur la rétine du capteur. Pour simplifier cette illustration, et surtout les suivantes, le plan de projection a été disposé devant le centre optique, en réalité il est derrière.

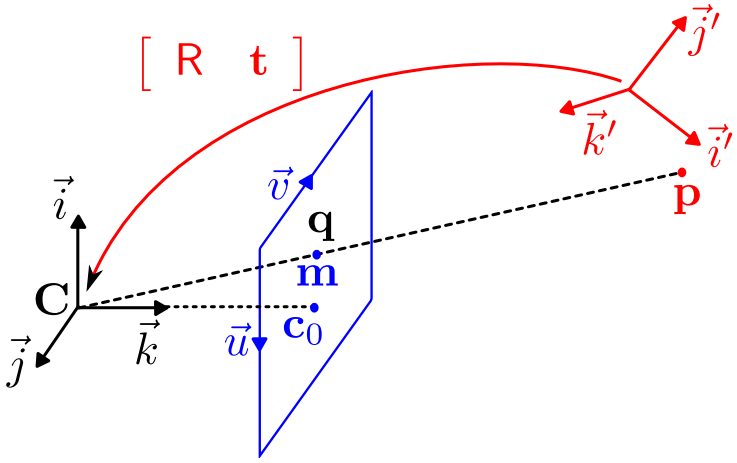


FIGURE II.12 – Modélisation de la projection perspective d'un point 3D $\mathbf{p}$, dont les coordonnées sont exprimées dans un repère monde (en rouge), sur le point $\mathbf{q}$ du plan image (en bleu). La matrice de changement de repère entre le repère monde et le repère caméra (en noir) doit être connue.

II.3 Étalonnage et calcul de pose

Dans la section II.2 il a été vu comment se modélisait la projection d'un point de la scène dans l'image grâce à la matrice de projection. Pour déterminer cette matrice, il existe de multiples méthodes selon que la matrice d'étalonnage soit connue ou non (Hartley and Zisserman, 2003; Forsyth and Ponce, 2003; Paragios et al., 2006). Lorsque la matrice d'étalonnage intrinsèque est connue, nous parlons alors plus précisément de calcul de pose (Dementhon and Davis, 1995; Quan, 1999; Ameller et al., 2002; Lepetit et al., 2009). Dans tous les cas il est nécessaire de connaître un modèle tridimensionnel de la scène avec des primitives, qui peuvent être des points, des droites ou des formes, que nous pourrons mettre en correspondance avec leurs images. Pour déterminer les paramètres manquants de la matrice de projection, il faut s'intéresser alors à l'erreur entre la position mesurée de la projection et celle prédite par le modèle de projection, c'est l'erreur de reprojection.

II.3.1 Estimation de la matrice de projection

En supposant que nous connaissions au minimum six points de la scène $\mathbf{p}_i, i \in [1, n], n \geq 6$ et les images $\mathbf{m}_i$ de cinq points et demi [5] par la projection, il est possible d'estimer la matrice de projection M définie précédemment (Hartley and Zisserman, 2003; Forsyth and Ponce, 2003; Paragios et al., 2006).

Cette estimation fait appel à la résolution d'un système linéaire obtenu à partir de l'équation (II.9). En notant $\mathbf{m}_i = \begin{bmatrix} u_i & v_i \end{bmatrix}^\top$ et ${\mathbf{l}_1}^\top, {\mathbf{l}_2}^\top$ et ${\mathbf{l}_3}^\top$ les trois lignes de M, nous pouvons écrire pour chaque point :

5. Le demi indique que nous avons besoin que de l'une ou l'autre des coordonnées de l'image de ce dernier point, mais bien de toutes ses coordonnées dans la scène.

$$\tilde{\mathbf{m}}_i = \begin{bmatrix} \mathbf{l}_1{}^\top \\ \mathbf{l}_2{}^\top \\ \mathbf{l}_3{}^\top \end{bmatrix} \tilde{\mathbf{p}}_i \tag{II.10}$$

$$\Leftrightarrow \mathbf{m}_i = \begin{bmatrix} \frac{\mathbf{l}_1{}^\top \tilde{\mathbf{p}}_i}{\mathbf{l}_3{}^\top \tilde{\mathbf{p}}_i} \\ \frac{\mathbf{l}_2{}^\top \tilde{\mathbf{p}}_i}{\mathbf{l}_3{}^\top \tilde{\mathbf{p}}_i} \end{bmatrix} \tag{II.11}$$

$$\Leftrightarrow \begin{bmatrix} u_i \mathbf{l}_3{}^\top - \mathbf{l}_1{}^\top \\ v_i \mathbf{l}_3{}^\top - \mathbf{l}_2{}^\top \end{bmatrix} \tilde{\mathbf{p}}_i = \mathbf{0} \tag{II.12}$$

$$\Leftrightarrow \underbrace{\begin{bmatrix} -\tilde{\mathbf{p}}_i^\top & \mathbf{0}^\top & u_i \tilde{\mathbf{p}}_i^\top \\ \mathbf{0}^\top & -\tilde{\mathbf{p}}_i^\top & v_i \tilde{\mathbf{p}}_i^\top \end{bmatrix}}_{=\mathsf{A}_i} \begin{bmatrix} \mathbf{l}_1 \\ \mathbf{l}_2 \\ \mathbf{l}_3 \end{bmatrix} = \mathbf{0}. \tag{II.13}$$

Pour les n points connus, nous empilons verticalement les matrices $\mathsf{A}_1, \ldots, \mathsf{A}_n$, toutes de taille 2×12, afin d'obtenir le système linéaire à résoudre sous la forme matricielle suivante :

$$\underbrace{\begin{bmatrix} \mathsf{A}_1 \\ \vdots \\ \mathsf{A}_n \end{bmatrix}}_{=\mathsf{A}} \underbrace{\begin{bmatrix} \mathbf{l}_1 \\ \mathbf{l}_2 \\ \mathbf{l}_3 \end{bmatrix}}_{=\mathbf{L}} = \mathbf{0}. \tag{II.14}$$

Lorsque il y a exactement cinq points et demi, la résolution peut être réalisée de manière exacte en utilisant une décomposition LU de la matrice A, méthode qui est présentée en annexe. Si plus de points sont présents, ce système est sur-déterminé et n'a plus de solution exacte. La résolution se réalise alors en minimisant la norme deux du vecteur $\mathsf{A}\mathbf{L}$. La méthode des moindres carrés linéaires permettant cette résolution sera étudiée section A.1. Il est à noter que le résultat retourné est normalisé c'est-à-dire que pour la solution trouvée $\|\mathbf{L}\| = 1$. La matrice de projection est donc obtenue à un facteur d'échelle près.

Dans le cas où les points de la scène utilisés pour l'estimation de la matrice de projection seraient tous coplanaires mais non alignés, alors la matrice A devient singulière puisqu'elle contiendra une colonne qui peut être nulle en considérant que le plan contenant les points a sa troisième coordonnée nulle. Il est toujours possible d'utiliser cette technique, mais il faut alors supprimer la troisième colonne de M pour lever la singularité de A. Cela ne nécessite plus que la connaissance de quatre points puisque la matrice estimée n'a plus que huit degrés de liberté contre onze précédemment, c'est donc une homographie.

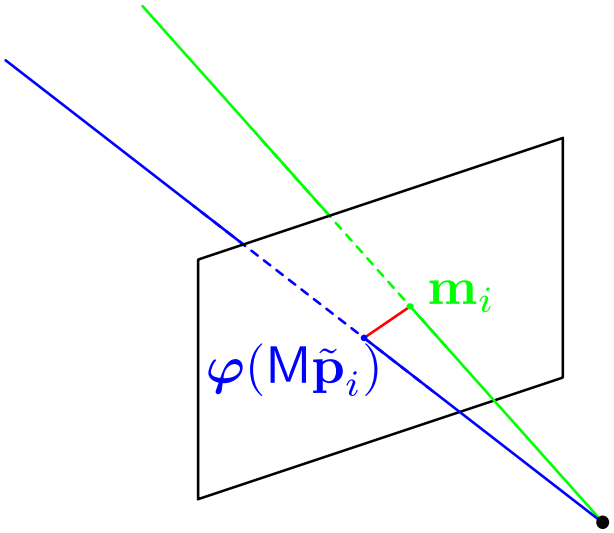


FIGURE II.13 – L'erreur de reprojection, ici en rouge, est la distance entre la projection mesurée d'un point $\mathbf{m}_i$, et sa projection estimée à partir des paramètres courants du modèle $\varphi(\mathsf{M}\tilde{\mathbf{p}}_i)$.

Pour obtenir des résultats stables numériquement il est nécessaire de normaliser les données d'entrée, c'est-à-dire les coordonnées des points dans l'image et les coordonnées des points de la scène. Faute de cela, la matrice A est souvent mal conditionnée dû aux écarts d'échelle entre ces différentes valeurs.

II.3.2 L'erreur de reprojection

Il est possible de raffiner la matrice de projection obtenue précédemment en optimisant ses paramètres. La fonction optimisée est constituée de l'erreur entre les points obtenus par la reprojection avec la matrice de projection et les points mesurés dans l'image. C'est l'erreur de reprojection géométrique qui est illustrée dans la figure II.13 pour un point. En considérant l'ensemble des points, elle s'exprime sous la forme

$$\sum_{i=1}^{n} \left(\mathbf{m}_i - \varphi(\mathsf{M}\tilde{\mathbf{p}}_i)\right)^2 . \tag{II.15}$$

La reprojection étant une fonction non-linéaire des paramètres de pose, cette optimisation fait appel à des méthodes spécifiques comme Levenberg-Marquardt qui seront abordées section A.3.

II.3.3 Étalonnage intrinsèque

Une fois la matrice de projection obtenue, il est possible d'extraire la matrice d'étalonnage K, ainsi que la pose de la caméra $\begin{bmatrix} \mathsf{R} & \mathbf{t} \end{bmatrix}$. Plusieurs méthodes permettent de réaliser ces calculs. La plus simple fonctionne par identification de la matrice $\beta\mathsf{M}$ avec $\mathsf{K}\begin{bmatrix} \mathsf{R} & \mathbf{t} \end{bmatrix}$ en utilisant les contraintes d'orthogonalité sur R, et où β est un coefficient qui permet de prendre en compte le fait que la matrice M a été obtenue à un coefficient près. Dans la littérature, il existe d'autres méthodes pour estimer K en s'intéressant par exemple à l'image de la conique absolue $(\mathsf{K}\mathsf{K}^\top)^{-1}$ comme dans (Sturm and Maybank, 1999) ou (Zhang, 1999). Nous avons utilisé la boîte-à-outils de (Bouguet, 2010) qui est basée en grande partie sur cette dernière méthode.

Des méthodes d'étalonnage basées sur d'autres objets connus comme des ellipses ou des lignes peuvent aussi être utilisées. Enfin il existe des méthodes d'auto-étalonnage (Triggs, 1997, 1998) qui utilisent plusieurs images d'un objet dont la structure tridimensionnelle n'est pas connue et est estimée en même temps que l'étalonnage.

II.3.4 Calcul de pose

Lorsque la matrice d'étalonnage est connue, le problème qui consiste à estimer la pose $\begin{bmatrix} \mathsf{R} & \mathbf{t} \end{bmatrix}$ de la caméra par rapport à un objet dans la scène a été très étudié (Dementhon and Davis, 1995; Lepetit et al., 2009). C'est un des problèmes fondamentaux en vision par ordinateur, souvent nommé «Perspective-n-Points» (PnP).

Les premières méthodes sont apparues à la fin des années 1980. Basées soit sur le calcul de la matrice de projection comme dans la section II.3.1, soit sur l'optimisation de l'erreur de reprojection, elles présentaient de nombreux inconvénients : manque de précision ou lenteur. Dans les années 1990 (Dementhon and Davis, 1995) proposent une nouvelle méthode reposant sur une approximation de la projection perspective qui est raffinée itérativement jusqu'à obtenir la projection perspective.

D'autres méthodes faisant appel à la résolution de plusieurs systèmes linéaires ont été proposées depuis, comme (Quan, 1999) ou encore (Ameller et al., 2002). Plus récemment, (Lepetit et al., 2009) proposent une méthode nommée «Efficient-PnP» qui s'exécute en temps linéaire et est très précise et robuste. Dans cette méthode la pose est paramétrée par quatre points de contrôle dont les positions sont déterminées par la résolution d'un système linéaire. Les paramètres de la pose sont ensuite extraits de la position de ces points de contrôle.

II.3.5 Distortion radiale et tangentielle

Comme vu dans la section II.1.2, lorsque les lentilles de l'objectif sont mal ajustées ou qu'un grand angle est utilisé, il peut se produire des phénomènes de distorsion géométrique dans l'image. Ces distorsions peuvent être modélisées (Fryer and Brown, 1986) afin d'être corrigées. Elles sont généralement plus importantes sur les bords de l'image qu'au centre, et sont donc modélisées pour chaque pixel en fonction de la distance à un centre de distorsion qui n'est pas nécessairement le point principal de l'image (la projection du centre optique) même si il est usuellement proche.

La correction de ces distorsions est relativement importante avant de réaliser un calcul de pose car leur présence fausse les résultats, tout particulièrement lorsque ces distorsions sont fortement marquées. À partir du moment où le nombre de points utilisés pour l'estimation de la matrice d'étalonnage est suffisant, l'estimation des paramètres de la distorsion (centre et force de la distorsion) peut être réalisée en même temps (Heikkilä and Silvén, 1997; Zhang, 1999). La distorsion radiale peut alors être supprimée en déplaçant chaque pixel sur sa position dans l'image si il n'y avait pas eu de distorsion.

Le modèle de distorsion qui est adopté par (Heikkilä and Silvén, 1997) et que nous avons donc utilisé exprime les coordonnées $\mathbf{q}''$ obtenues après les distorsions selon

$$\mathbf{q}'' = (1 + c_1 r^2 + c_2 r^4 + c_5 r^6)\mathbf{q}' + \mathbf{d}_\mathbf{q}, \quad \mathbf{d}_\mathbf{q} = \begin{bmatrix} 2c_3 ab + c_4(r^2 + 2a^2) \\ c_3(r^2 + 2b^2) + 2c_4 ab \end{bmatrix}, \tag{II.16}$$

où $r^2 = a^2 + b^2$ est la norme au carré de $\mathbf{q}'$ et $\begin{bmatrix} c_1 & \dots & c_5 \end{bmatrix}^\top$ est un vecteur des cinq paramètres utilisés pour modéliser les distorsions. Il est estimé lors de l'étalonnage. Connaissant ce vecteur de paramètres, il est aisé de retrouver $\mathbf{q}'$ à partir de $\mathbf{q}''$, c'est pourquoi par la suite, nous considérerons toujours les coordonnées comme ayant été corrigées. Un exemple de correction d'une image comportant de la distorsion radiale est donné dans la figure II.14.

(a) Image originale. (b) Image corrigée.

FIGURE II.14 – Exemple de correction d'une image comportant de la distorsion radiale. Image de Gabriel Peyrot (http://gabrielpeyrot.com/).

II.4 Mise en correspondance d'images

Dans les méthodes d'étalonnage et de calcul de pose présentées précédemment, il est nécessaire de connaître les correspondances entre les points tridimensionnels de la scène et leurs projections dans l'image. Il est toujours possible de construire ces correspondances manuellement, néanmoins cela demande du temps.

Lorsque le but est de réaliser l'étalonnage d'une caméra, ce processus est bien souvent partiellement automatisé par l'utilisation d'une mire contenant un damier comme présentée figure II.15. Les quatre coins extérieurs de la mire sont alors généralement pointés manuellement puis les autres coins sont détectés automatiquement à l'intérieur du polygone défini par ces quatre coins en recherchant les intersections des bordures du damier par une analyse du gradient de l'intensité lumineuse dans l'image (Shi and Tomasi, 1994). C'est ce qui est réalisé dans la boîte-à-outils de (Bouguet, 2010) que nous avons utilisée.

Dans le cas d'un objet suffisamment texturé, une mise en correspondance automatique sera réalisée entre les projections d'un même point de l'objet dans les différentes images. Ces points sont appelés «points d'intérêt» et peuvent être détectés automatiquement (Lowe, 2004; Bay et al., 2006). Les méthodes de détection automatique se basent principalement sur l'analyse des variations de grande amplitude du gradient de l'image, un objet contiendra donc d'autant plus de points d'intérêt qu'il contiendra de nombreux contours. La figure II.16 montre un exemple d'objet contenant peu de points d'intérêt car les variations de gradient sont de faible amplitude, et un autre en contenant davantage.

Afin de réaliser la mise en correspondance des points d'intérêt détectés, nous leur associons des descripteurs comme (Lowe, 2004; Bay et al., 2006). Ces derniers caractérisent les points d'intérêt en se basant sur l'analyse de leur voisinage à plus ou moins grande échelle. Cela permet de définir une mesure de la ressemblance entre deux points d'intérêt qui est ensuite utilisée pour les mettre en correspondance. Il est nécessaire que les voisinages des points soient suffisamment riches en information et différents les uns des autres pour pouvoir être identifiés clairement. Par exemple, la figure II.17 montre un objet qui contient de nombreux points d'intérêt, mais qui ne sont pas discernables automatiquement pour deux raisons : la texture est répétée par endroits, les points d'intérêt se ressemblent tous beaucoup, leurs voisinages étant peu différents.

Nous avons utilisé l'implémentation libre du descripteur SIFT (Lowe, 2004) fournie dans (Vedaldi and Fulkerson, 2008), car cette méthode est robuste aux changements d'échelle, d'illumination et de point de vue. La mise en correspondance de deux points d'intérêt, et donc de leurs deux descripteurs, est réalisée par une méthode du type «Winner-take-all» (Scharstein and Szeliski, 2002). Le principe est d'attribuer un score à chaque candidat à la correspondance en fonction de la similarité du descripteur du point considéré et du candidat. Le gagnant est sélectionné comme celui ayant le meilleur score.

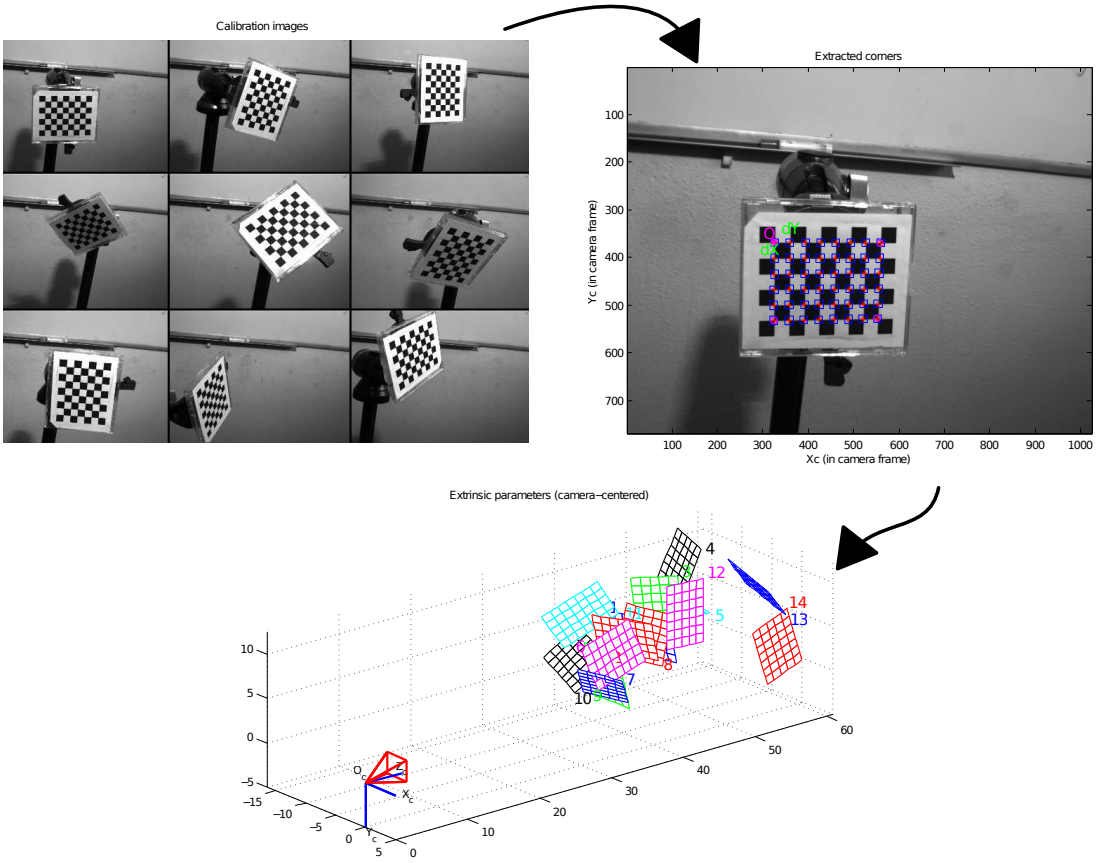
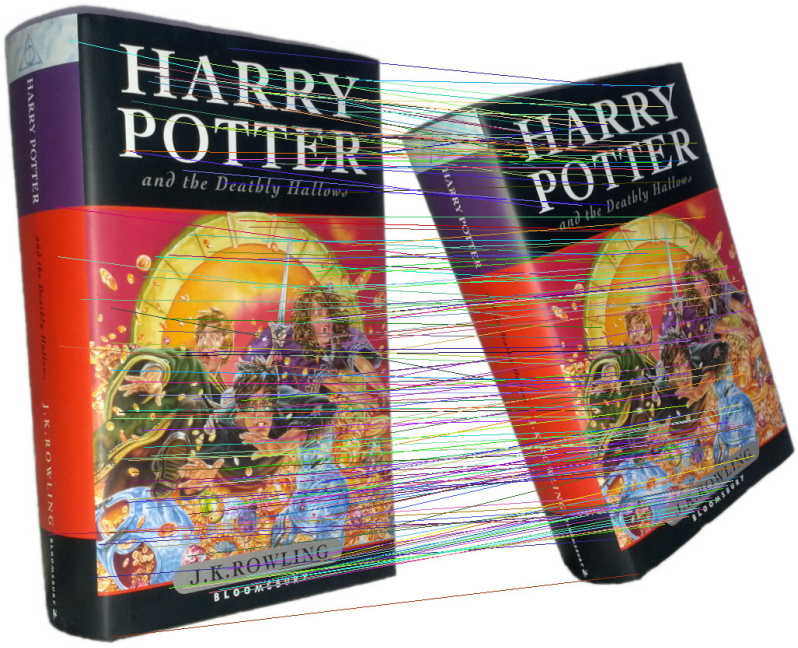


FIGURE II.15 – L'étalonnage à partir d'une mire en damier se réalise en trois étapes. Tout d'abord il faut acquérir plusieurs images de la mire qui sont ensuite traitées pour extraire de manière semi-automatique les coins du damier. Enfin l'estimation des différentes matrices de projection permet de retrouver la matrice d'étalonnage (ainsi que les différentes poses de la caméra ou de l'objet).

(a) Objet contenant peu de points d'intérêt car peu de variations de gradient.

(b) Objet contenant davantage de contours et donc de points d'intérêt.

FIGURE II.16 – L'impact des variations du gradient sur les détecteurs automatiques de points d'intérêt est important. Le manque de contours facilement détectables, et provoquant donc un fort gradient, se fait nettement ressentir avec le premier objet. Ces images ont été obtenues avec le détecteur SIFT (Lowe, 2004; Vedaldi and Fulkerson, 2008).



(a) Objet très texturé, les correspondances sont quasiment toutes justes et forment des lignes globalement toutes dans la même direction.

(b) Objet insuffisament texturé, de nombreuses correspondances sont erronées.

FIGURE II.17 – Plus la texture est riche, et donc plus le voisinage à plus ou moins grande échelle des points d'intérêt est varié, meilleures sont les mises en correspondances.

Dans l'implémentation utilisée, cette méthode est adaptée pour rejeter le gagnant lorsque la différence de score entre le gagnant et le deuxième est trop faible. Cela permet d'éliminer les correspondances trop ambiguës. Ce principe est souvent nommé «Lowe Ratio Test» puisqu'il a été introduit par (Lowe, 2004).

CHAPITRE III

LE «ROLLING SHUTTER»

Résumé

Le «Rolling Shutter» trouve son origine dans la technologie employée au sein des capteurs pour créer puis récolter les charges électriques des cellules photosensibles touchées par les rayons lumineux. En première partie nous étudierons et comparerons les deux technologies actuellement utilisées puis définirons exactement ce qu'est le «Rolling Shutter» et quelle est son origine. Les différentes conséquences sur l'image que peut avoir ce mode d'acquisition seront explorées en détails en deuxième partie. Enfin en troisième partie la littérature existante sur le «Rolling Shutter» sera étudiée, tant sur le plan de la correction de ces conséquences que de leur utilisation comme capteur de mouvement.

III.1 Origine du «Rolling Shutter»

Il existe à ce jour deux grandes technologies pour récolter les charges des cellules photosensibles d'un capteur de caméra. Il s'agit des technologies «Charge-Coupled Device» (CCD) et «Complementary Metal Oxide Semiconductor» (CMOS), les deux ayant des avantages et des inconvénients.

III.1.1 Notion d'exposition et d'intégration

Quelle que soit la technologie employée, l'obtention de l'image par le capteur se décompose en deux étapes importantes : l'exposition de la rétine et la récolte des charges électriques. L'ordonnancement de ces deux étapes au sein du capteur est la clef de la compréhension des différences entre les deux technologies qui sont présentées par la suite, mais surtout c'est lui qui est à l'origine de l'effet étudié.

L'exposition de la rétine du capteur se déroule de la manière suivante : dans un premier temps, les cellules photosensibles sont déchargées de leurs charges électriques résiduelles. Dans un deuxième temps, les cellules photosensibles sont laissées un certain temps, appelé temps d'exposition, exposées à l'image optique issue de l'objectif de la caméra. Les cellules s'imprègnent ainsi de la luminosité de cette image optique et accumulent une charge électrique qui y est proportionnelle. Le temps d'exposition est une propriété paramétrable des caméras qui permet d'ajuster la quantité de lumière reçue par le capteur et donc la luminosité de l'image finale. Pour chaque capteur, il existe un intervalle dans lequel ce temps doit se trouver : trop faible, il ne permet pas de recevoir suffisamment de lumière pour récolter autre chose que du bruit, et trop élevé les cellules photosensibles se retrouvent généralement saturées. Pour des utilisation classiques, un temps d'exposition compris entre cent micro secondes et une seconde est généralement utilisé.

Une fois le temps d'exposition écoulé, nous procèdons à l'intégration : les cellules sont à nouveau déchargées de leurs charges électriques par un circuit électronique constitué d'un amplificateur opérationnel. Le résultat est une tension qui est ensuite échantillonnée pour être transformée en signal numérique. Le temps nécessaire à ce traitement est appelé temps d'intégration, et il dépend fortement de la technologie employée dans le capteur. En effet, l'ensemble de cette étape peut se dérouler de différentes manières selon le circuit électronique utilisé pour traiter les charges électriques.

Après l'intégration, une dernière étape est présente et consiste à extraire le signal numérique à la sortie du capteur pour le délivrer ou transférer au système auquel est connectée la caméra selon un format standardisé. C'est un circuit de la caméra mais extérieur au capteur qui prend en charge cette opération, et bien que le temps nécessaire à celle-ci n'influence pas sur le temps nécessaire pour obtenir une seule image, il influe sur le temps entre l'obtention de deux images par le système. Cette étape est appelée normalement acquisition, mais ce terme est également bien souvent employé pour désigner

l'ensemble du processus d'obtention de l'image par le système utilisateur de la caméra, c'est-à-dire exposition, intégration et acquisition.

III.1.2 Technologie CCD

Avec la technologie CCD, la rétine est d'abord remise intégralement à zéro puis exposée en une seule fois à la scène durant le temps d'exposition. Une fois la rétine exposée, l'intégration se réalise elle aussi pour l'ensemble des cellules. Dans un circuit CCD, il n'y a généralement qu'un seul amplificateur-convertisseur analogique/numérique qui réalise l'intégration. De ce fait, avec cette technologie, l'acquisition est nécessairement réalisée pour une image entière.

Le temps d'intégration est par conséquent élevé et incompressible puisqu'il faut transférer et convertir les charges de toutes les cellules une par une. Pour cela, au bord du capteur se trouve une colonne supplémentaire qui n'est pas exposée à la scène et au bout de laquelle ce trouve l'amplificateur-convertisseur. Les charges sont transférées colonne par colonne sur cette colonne temporaire où elles sont ensuite transférées ligne par ligne dans l'amplificateur-convertisseur. Ce mécanisme est illustré dans la figure III.1.

Sur les capteurs CCD récents et afin d'accélérer le processus, des colonnes de stockage temporaire se trouvent au sein de la rétine entre chaque colonne de cellules photosensibles exposées à la scène. À la fin de l'exposition, les charges des cellules photosensibles sont transférées sur cet espace de stockage avant l'intégration des différentes colonnes. Cela permet de réaliser l'exposition des cellules photosensibles pour l'image suivante en même temps que se fait l'intégration de l'image courante, réduisant ainsi le temps nécessaire entre l'obtention de deux images. Cependant, il est toujours nécessaire de traiter l'ensemble des cellules et le temps d'intégration étant généralement nettement supérieur au temps d'exposition, le gain en vitesse reste faible. Avec ces dernières avancées, les capteurs CCD les plus rapides peuvent fournir entre cent et deux cents images par secondes dans les résolutions standards.

III.1.3 Technologie CMOS

Pour la technologie CMOS, à chaque cellule photosensible est accolé directement un amplificateur-convertisseur. La remise à zéro et l'exposition ne se font donc plus en une seule fois pour l'ensemble de la rétine mais à la demande pour une ligne donnée. Après l'exposition de la ligne, le temps d'intégration est réduit puisque toutes les cellules de cette ligne amplifient et convertissent leurs charges en même temps. Il ne reste alors qu'à lire le signal résultant sur les colonnes souhaitées. Nous pouvons ainsi accéder en lecture à n'importe quelle cellule de manière spécifique et ne réaliser l'acquisition que de certaines parties de l'image.

Un moyen très rapide d'obtenir toute ou partie de l'image est d'exposer les lignes choisies les unes

après les autres comme illustré dans la figure III.2. Ce mode de fonctionnement particulier est appelé «Rolling Shutter» et est propre à la technologie CMOS. Il est extrêmement rapide, car l'intégration d'une ligne est réalisée en un temps très court et durant lequel la ligne suivante peut déjà être exposée. Le gain en temps entre l'obtention de deux images complètes est alors significatif, nous pouvons obtenir des centaines d'images par seconde dans les résolutions standards. Il est encore plus élevé lorsque seulement une petite partie de l'image est nécessaire, des vitesses pouvant aller jusqu'à plusieurs milliers d'images par seconde sont alors atteignables.

III.1.4 Avantages et inconvénients

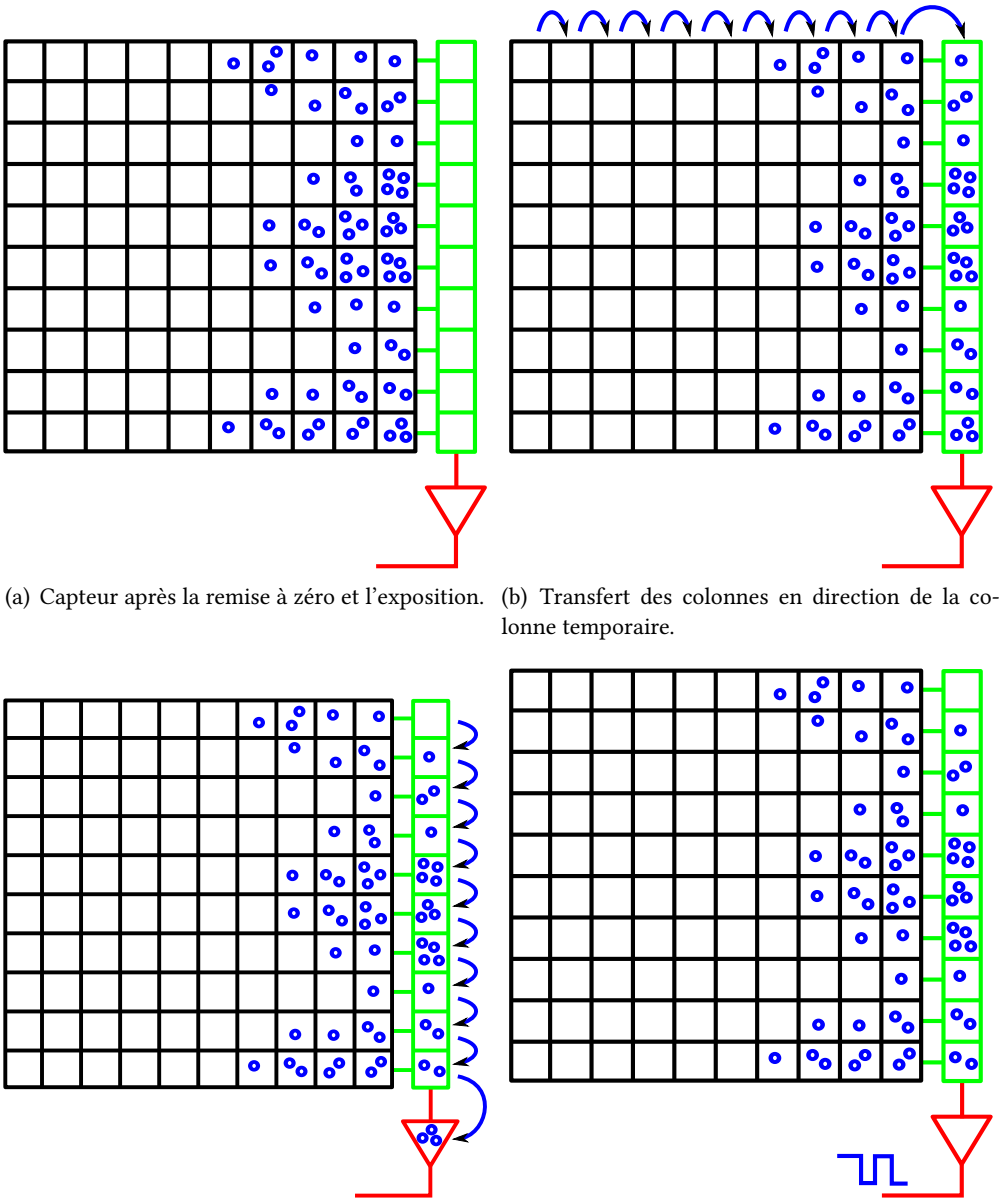
Chacune des deux technologies présente des avantages et des inconvénients. Tout d'abord comme il est possible de le constater dans les figures III.1 et III.2, la place sur le capteur dédiée à la cellule photosensible est nettement réduite avec la technologie CMOS, c'est ce qui s'appelle le «fill factor». Les cellules étant plus petites, elles reçoivent donc moins de lumière et par conséquent il faut amplifier d'avantage le signal ce qui introduit plus de bruit dans l'image.

Néanmoins l'électronique nécessaire pour gérer l'intégration est réduite avec la technologie CMOS, ce qui donne un coût de production plus faible et une consommation électrique diminuée. De plus, le temps d'intégration étant nettement réduit avec la technologie CMOS, le nombre possible d'images capturées par seconde est donc supérieur à celui obtenu en utilisant la technologie CCD.

Enfin alors qu'avec la technologie CCD la rétine est nécessairement exposée en une seule fois, elle est exposée séquentiellement avec la technologie CMOS. Lorsque la scène est figée et qu'il n'y a pas de mouvement de la caméra le résultat est strictement identique, mais si ce n'est pas le cas la technologie CCD créé un flou dans l'image alors que la technologie CMOS produit des déformations dans l'image. Ce phénomène sera étudié en détail dans la section suivante.

La table présentée figure III.3 récapitule les avantages et inconvénients de ces deux technologies.

Les capteurs CCD sont de plus en plus souvent réservés à des applications haut de gammes, tandis que les capteurs CMOS envahissent les caméras des nombreux périphériques mobiles d'aujourd'hui.



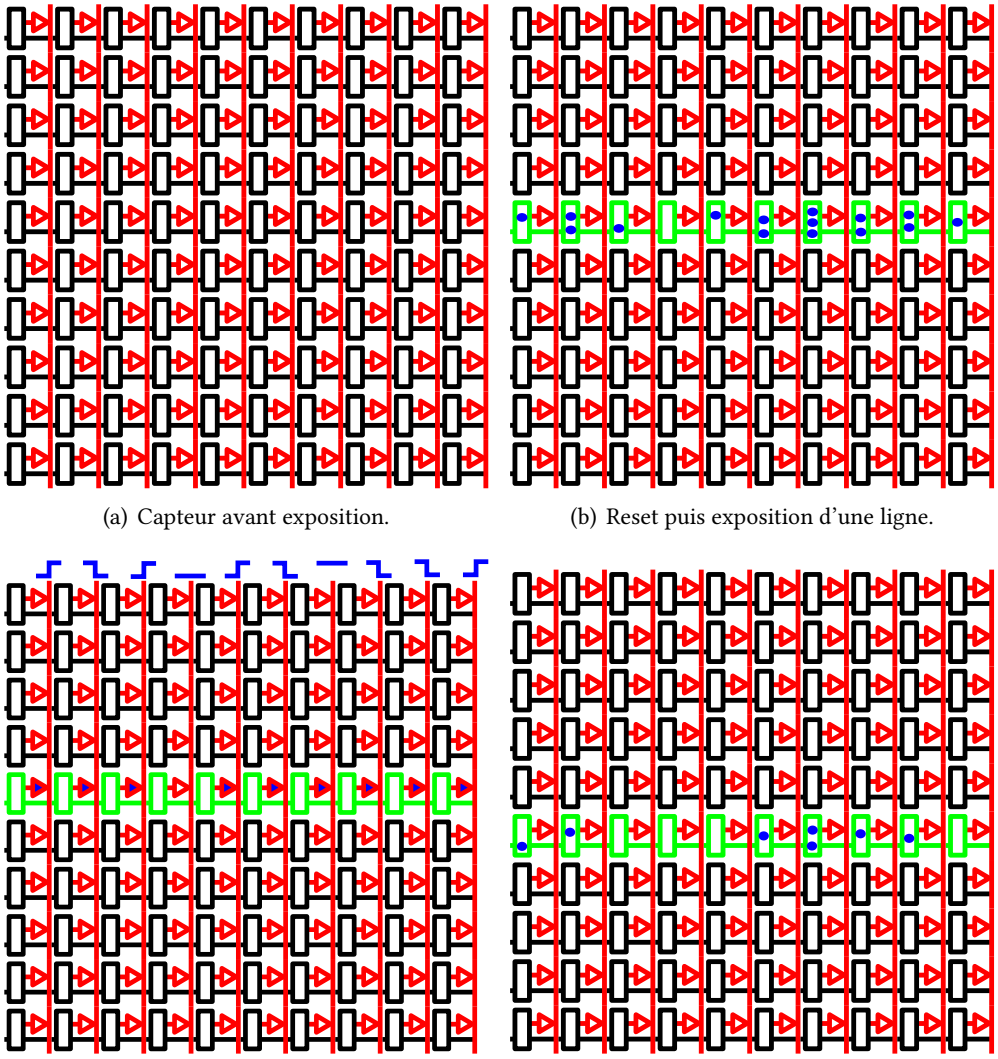
(a) Capteur après la remise à zéro et l'exposition.

(b) Transfert des colonnes en direction de la colonne temporaire.

(c) Transfert des lignes de la colonne temporaire vers l'amplificateur.

(d) Amplification et conversion.

FIGURE III.1 – L'exposition et l'intégration d'une image avec un capteur de technologie CCD : une fois la rétine remise à zéro puis exposée à la lumière de la scène (a), les charges sont transférées de colonnes en colonnes jusqu'à la colonne temporaire en vert (b). Une fois sur cette colonne, les charges sont transférées ligne par ligne (c) vers l'amplificateur-convertisseur qui délivre alors le signal numérique correspondant (d).



(a) Capteur avant exposition.

(b) Reset puis exposition d'une ligne.

(c) Amplification et conversion de la ligne.

(d) Exposition de la ligne suivante.

FIGURE III.2 – Avec un capteur de technologie CMOS, chaque cellule de la rétine dispose de son propre amplificateur (a). L'exposition et l'intégration se font ainsi indépendamment ligne par ligne. La ligne choisie est remise à zéro puis exposée à la lumière de la scène (b). Toutes ses colonnes sont ensuite amplifiées et converties en signal électronique en même temps (c). Pendant ce temps nous pouvons déjà réaliser la mise à zéro et l'exposition de la ligne suivante (d).

	Acquisition	Transfert de charge	Mouvement	Luminosité / Sensibilité	Coût	Énergie
CCD	Rétine entière	Colonnes à colonnes	Flou	Grande	Élevé	Élevée
CMOS RS	Une ligne ou ROI	Aucun	Déformation	Faible	Faible	Faible

FIGURE III.3 – Alors que la technologie CCD a longtemps dominée le marché des capteurs de caméras en raison de sa meilleure sensibilité et de son meilleur ratio signal sur bruit, cette technologie tend désormais à être réservées aux applications haut de gamme (cinéna par exemple). La technologie CMOS est en plein essor, notamment dans les applications mobiles embarquées où elle ne présente que des avantages.

III.2 Les effets du «Rolling Shutter»

Cette section explore les différents effets dans l'image du mode d'acquisition «Rolling Shutter» des caméras comportant des capteurs CMOS lorsque la scène est dynamique ou que la caméra se déplace. Alors que certains mouvements ne causent que de simples déformations des objets filmés, d'autres peuvent entraîner des occultations ou des répétitions de points dans l'image. Les conséquences néfastes du «Rolling Shutter» sur l'estimation de pose classique sont aussi étudiées.

III.2.1 Conséquences du mouvement

Bien que le mouvement des objets ou de la caméra soit généralement complexe, il peut être décomposé en un ou plusieurs mouvements basiques qui forment les distortions observées. Il s'agit des translations horizontales, verticales ou le long de l'axe optique qui ont chacune un effet propre. Tout autre mouvement, et en particulier les rotations, entraîne des déformations dans l'image qui peuvent s'expliquer par l'effet de ces mouvements de base.

III.2.1.1 Translations horizontales

Lorsque la caméra ou un objet de la scène se déplace le long de l'axe horizontal de la rétine, il se produit une déformation de la scène ou de l'objet en mouvement qui tend à courber ou incliner cette partie de l'image comme illustré dans la figure III.4. Si le mouvement est uniforme, cela revient simplement à pencher l'objet ou la scène plus ou moins selon la vitesse du mouvement. Le sens de l'inclinaison dépend du sens du mouvement relatif équivalent entre la scène ou l'objet et la caméra qui serait fixe.

Avec ce type de mouvement, tous les points de la scène ou de l'objet se retrouvent bien projetés dans l'image, ils sont simplement décalés horizontalement au fur et à mesure des lignes. Il n'y a donc dans ce cas pas de points manquants ou répétés. Un exemple de ces distorsions est fourni dans la figure III.5.

III.2.1.2 Translations verticales

Les translations le long de l'axe vertical de la rétine produisent un effet de compression ou d'étirement de l'image selon le sens du mouvement. L'effet de compression est obtenu lorsque le mouvement est en sens inverse par rapport au sens de parcours des lignes. Un exemple de chacun de ces deux effets est montré dans la figure III.6.

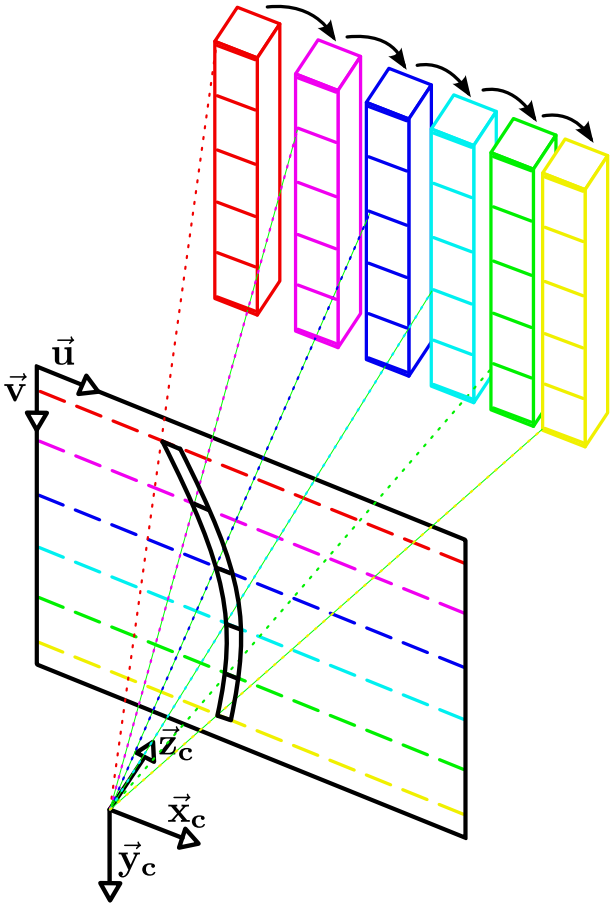


FIGURE III.4 – Lorsqu'un mouvement de l'objet parallèle à l'axe horizontal de la rétine du capteur se produit, les différents points de l'objet sont projetés avec un décalage horizontal entre chaque lignes.

(a) Translation vers la gauche de la caméra.

(b) Vue avec la caméra fixe.

(c) Translation vers la droite de la caméra.

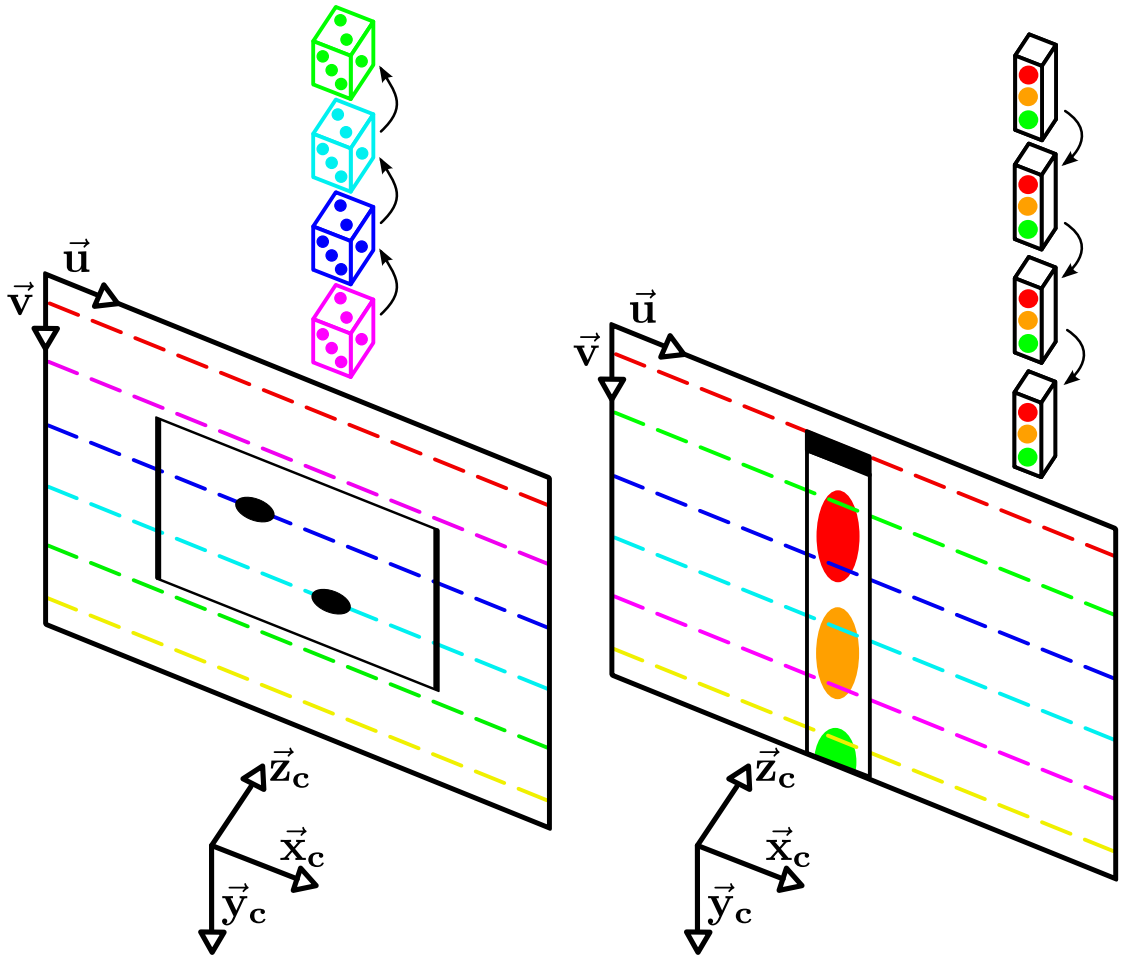
FIGURE III.5 – Exemple d'images obtenues lors d'un mouvement de la caméra le long de l'axe horizontal de la rétine. Le bâtiment semble pencher d'un côté ou de l'autre en fonction du sens du mouvement.

(a) Translation vers le bas de la caméra : compression de l'image de l'objet.

(b) Vue avec la caméra fixe.

(c) Translation vers le haut de la caméra : étirement de l'image de l'objet.

FIGURE III.6 – Exemple d'images obtenues lors d'un mouvement de la caméra le long de l'axe vertical de la rétine. La scène se retrouve légèrement étirée ou compressée selon le sens du mouvement.



(a) Translation vers le haut de l'objet : compression de l'image de l'objet.

(b) Translation vers le bas de l'objet : étirement de l'image de l'objet.

FIGURE III.7 – Lorsqu'un mouvement de l'objet parallèle à l'axe vertical de la rétine se produit, l'objet s'étire ou se compresse dans l'image. Pour simplifier cette illustration, un seul plan de l'objet est projeté, il s'agit du plan fronto-parallèle à la rétine. Dans la première image, un dé est jeté en l'air devant la caméra statique, la face avant capturée par la caméra montre seulement deux points qui sont écrasés alors que le dé en comporte trois et ronds. Le troisième qui aurait dû être au centre est passé devant la caméra entre la capture de la ligne bleue et celle de la ligne cyan. Dans la seconde image, nous prenons une photo d'un feu statique tout en relevant l'appareil photo. Les lumières rouge et orange du feu apparaissent comme des ellipses allongées dans le sens vertical alors que la projection perspective d'un cercle fronto-parallèle reste un cercle.

Lorsque l'image est comprimée par le mouvement, des points de la scène peuvent disparaître de l'image si le mouvement est suffisamment rapide par rapport à la vitesse de parcours des lignes. Inversement lorsque l'image est étirée par le mouvement, certains points peuvent être répétés sur une ou plusieurs lignes. Ces deux effets sont montrés dans la figure III.7.

III.2.1.3 Translations sur la profondeur

Lorsque l'objet ou la scène se déplace le long de l'axe de profondeur de la caméra, de part la perspective sa projection tend soit à diminuer en taille lors d'un éloignement ou au contraire à grossir lors d'un rapprochement. L'effet obtenu avec un caméra «Rolling Shutter» est alors une déformation de l'objet en trapèze le long des lignes de l'image, la pointe du trapèze étant orientée selon le mouvement et le sens de balayage des lignes. Une illustration de cet effet est donnée figure III.8. Il est difficile à observer car la vitesse de translation nécessaire pour qu'il soit correctement perceptible est très élevée. Un exemple de légère déformation de ce type peut être observé figure III.9. Contrairement à l'effet présenté précédemment, celui-ci ne provoque pas de disparition ou de duplication des projections des points.

III.2.1.4 Rotations

Localement une rotation entraîne une translation tangentielle des points. L'effet obtenu est donc une combinaison des trois précédents, et dont le sens dépend du sens de la rotation et de la position du point dans la rotation. Un exemple est donné dans la figure III.10 où un objet en rotation plane a été filmé avec une caméra «Rolling Shutter». Les distorsions peuvent s'expliquer en considérant les quatre quarts de la rotation séparés par l'axe vertical et horizontal. Dans chacun de ces quarts, la distorsion est différente puisque l'orientation de la translation tangentielle change.

Comme cet effet est une combinaison des effets précédents, il faut bien noter que des points peuvent disparaître de l'image ou se répéter plusieurs fois. Dans des cas extrêmes de rotation très rapide, il peut même arriver qu'un point disparaisse de l'image à un endroit et réapparaisse à un ou plusieurs autres endroits. En présence de telles rotations, il peut donc arriver que des points donnent l'impression d'avoir changé d'ordre ou se répètent de manière périodique dans l'image. C'est ce qui peut être observé dans l'image III.11.

III.2.2 Estimation de pose en «Rolling Shutter»

L'utilisation des algorithmes classiques d'estimation de pose tels que EPnP ou la minimisation de l'erreur de reprojection sur des images comportant des distorsions dues au «Rolling Shutter» donne des résultats mitigés.

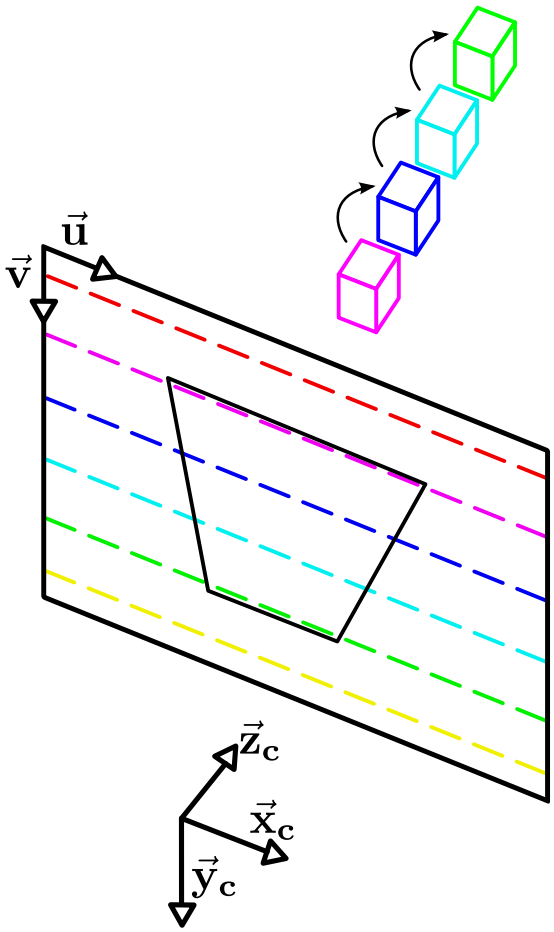
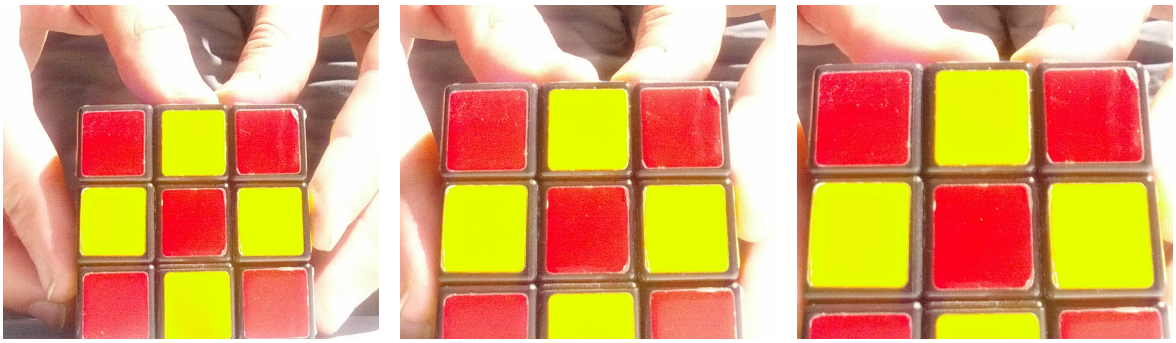


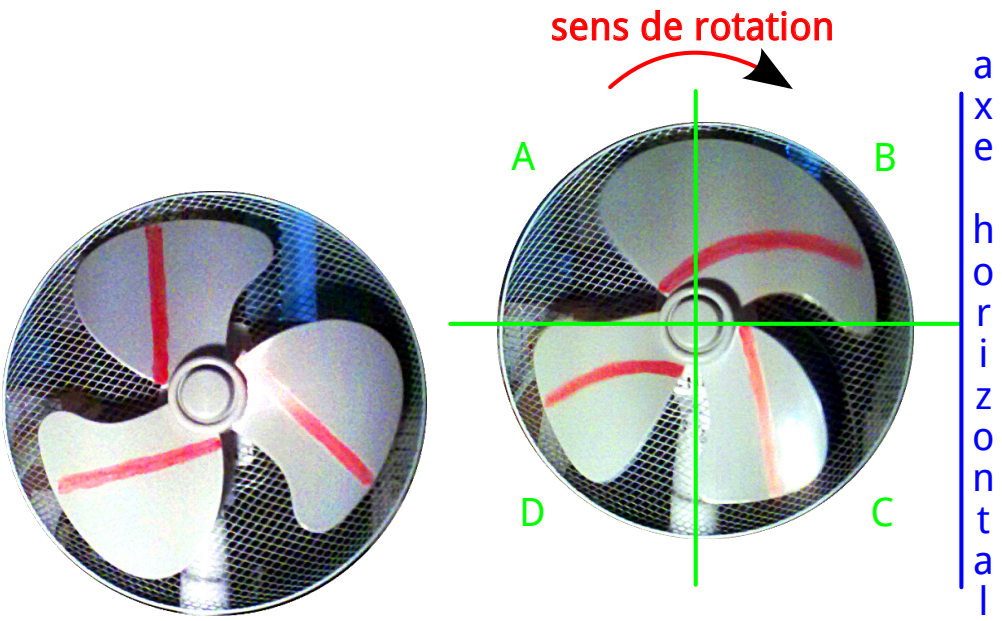
FIGURE III.8 – Une translation le long de l'axe de profondeur de la caméra induit une déformation en trapèze. Ici l'objet s'éloigne de la caméra, son image sur la rétine est de plus en plus petite.



(a) Objet fixe. (b) Début de la translation. (c) Fin de la translation.

FIGURE III.9 – Exemple d'images obtenues lors d'un mouvement de l'objet le long de l'axe de profondeur de la caméra. Nous observons sur les images (b) et (c) un légère déformation en trapèze de l'objet : lors de l'acquisition des dernières lignes de l'image, l'objet était plus proche que lors de l'acquisition des premières lignes.

Lorsque les déformations sont peu importantes, le résultat est souvent proche de la pose moyenne de l'objet ou de la caméra pendant l'exposition de l'image. En revanche lorsque les déformations sont suffisamment amples, il peut arriver que la pose retournée par les algorithmes classiques soit totalement erronée et sans rapport avec la configuration de la scène, comme cela peut être observé dans la figure III.12.



(a) Ventilateur fixe. (b) Ventilateur tournant

FIGURE III.10 – Exemple de distorsions dues à une rotation. Pour des raisons pratiques la caméra se situe verticalement, l'axe horizontal en bas de la rétine se situe donc verticalement à droite de l'image. Pour interpréter les déformations, il faut séparer la rotation en quadrants. Dans les quadrants A et B, la rotation entraîne localement une translation vers le bas de la rétine, d'où l'effet d'étirement de la pale du ventilateur. Dans les quadrants C et D, c'est l'inverse qui se produit, les pales y sont donc comprimées. Dans les quadrants B et C, la rotation entraîne aussi une translation le long de l'axe horizontal, raison pour laquelle le trait rouge se retrouve courbé. Dans le quadrant D, la translation horizontale se fait dans l'autre sens, c'est pourquoi le trait penche dans l'autre sens. Le trait rouge sur la pale qui se trouve dans le quadrant C est presque droit car il se situe non loin de l'endroit où la translation horizontale devient nulle.

FIGURE III.11 – Une rotation extrêmement rapide peut entraîner des disparitions et réapparitions de points de manière surprenante. Cette hélice d'avion n'a bien évidemment pas autant de pales que l'image en contient, et elles sont bien toutes attachées à l'avion. Photographié par Derek Sine.

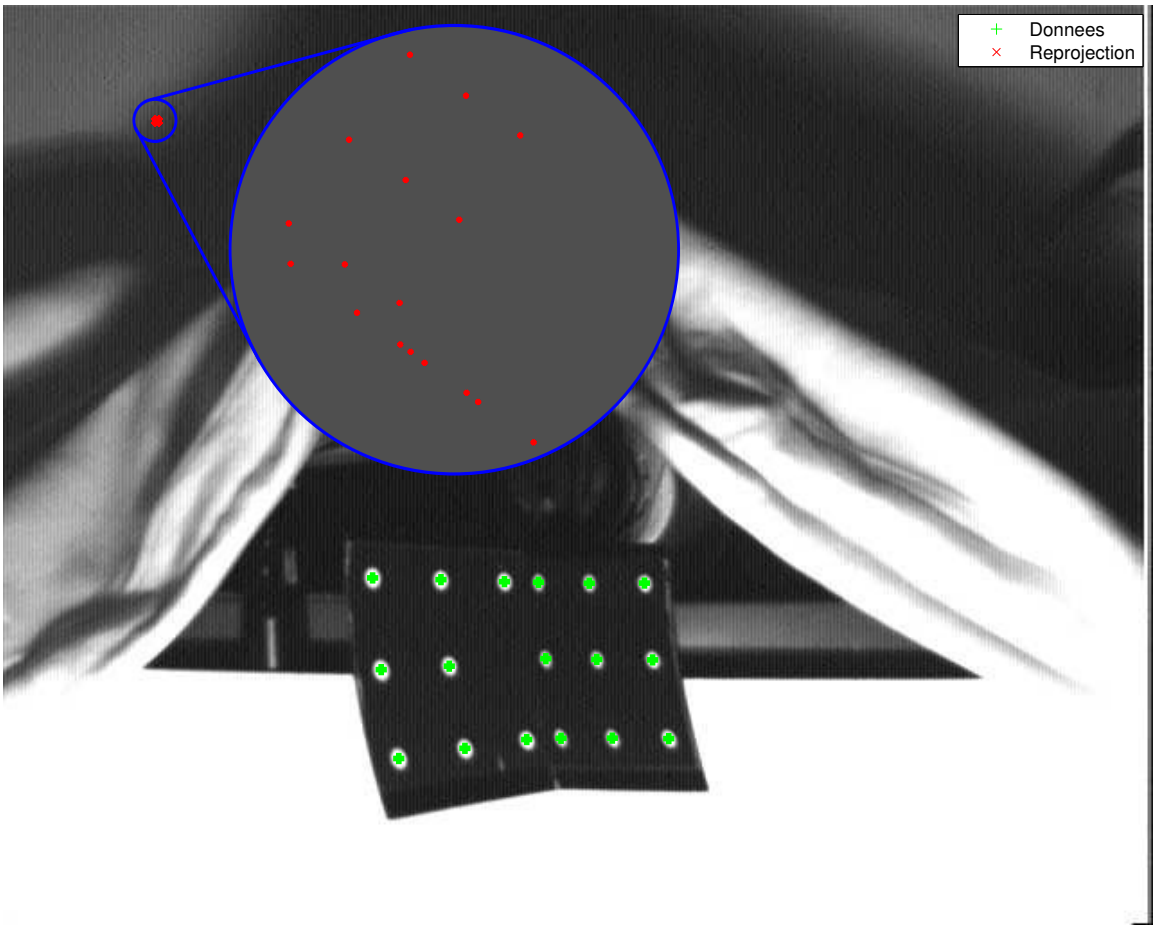


FIGURE III.12 – Exemple d'estimation de pose avec la méthode «EPnP» sur une image issue d'une caméra «Rolling Shutter». Les points de données verts sont les centres des pastilles blanche sur l'objet qui est en rotation rapide. Les points obtenus après reprojection avec la pose estimée sont en rouge.

III.3 État de l'art sur le «Rolling Shutter»

En raison de leur coût et consommation moindres que les capteurs CDD, nous trouvons désormais les capteurs CMOS dans de nombreux appareils (souvent mobiles) tels que appareils photographique et caméscopes bas de gamme, caméras d'ordinateurs portables ou de téléphones mobiles. Lors de l'utilisation habituelle de ces caméras, que ce soit pour faire un film souvenir des vacances, faire de la photographie d'événements sportifs,... les distorsions créées par le «Rolling Shutter» sont dérangeantes et produisent des résultats difficilement exploitables en l'état, notamment si le mouvement est rapide. La problématique de la correction de ces distorsions est donc apparue très tôt lorsque les capteurs CMOS ont commencé à se répandre massivement au début des années 2000. D'autres travaux se sont cependant attachés à modéliser l'effet du «Rolling Shutter» afin des l'exploiter pour en tirer de l'information utile, notamment pour des applications en robotique. En effet les distorsions étant le résultat du mouvement dans la scène ou de la caméra, elles contiennent donc de l'information sur ce mouvement.

III.3.1 Correction «Rolling Shutter»

III.3.1.1 Méthodes ad-hoc utilisant une séquence d'images

Plusieurs méthodes (Hwang, 2003; Im et al., 2006; Chun et al., 2008; Liang et al., 2008) sont rapidement apparues autour de l'idée consistant à utiliser le flot optique entre une image et la suivante pour estimer le mouvement dans cette dernière et ensuite transformer l'image pour le compenser en décalant simplement les lignes horizontalement. Alors que les premières approches (Hwang, 2003; Im et al., 2006) se contentaient d'estimer directement le mouvement par le flot optique, la dernière en date (Liang et al., 2008) estime les paramètres d'une courbe de bézier modélisant le mouvement pour apporter une meilleure correction. Ces méthodes calculent un mouvement global sur toute l'image et ne permettent donc pas de compenser un mouvement propre à certains objets de la scène.

Des méthodes alternatives (Cho et al., 2007; Cho and Hong, 2007, 2008; Hong et al., 2012) basées sur la correction de l'image en lui appliquant des transformations affines sur une grille de points ont donc été élaborées. Ces méthodes commencent généralement par l'estimation du mouvement global avec le flot optique et qui est ensuite raffiné localement sur les points de la grille. En conservant le modèle de (Cho and Hong, 2007), il a été montré dans (Cho and Kim, 2012) qu'il était possible de corriger les images issues de caméras «Rolling Shutter» afin de réaliser une image panoramique qui ne soit pas distordue.

Bien que difficile à comprendre avec les informations disponibles, la méthode présentée dans le brevet (D'Angelo and Vandergheynst, 2011) permet également de créer une image où l'effet «Rolling Shutter» est compensé à partir de deux images acquises en mode «Rolling Shutter». La plupart des

logiciels de traitement vidéo proposent des méthodes pour stabiliser les vidéos filmées en tenant la caméra manuellement. Certains de ces logiciels (ProDAD, 2010; Thalin, 2011) prennent en compte les effets du «Rolling Shutter» et tentent de les corriger, généralement en redressant les lignes verticales dans l'image.

Les méthodes présentées précédemment ont l'inconvénient de ne pas modéliser le mouvement tridimensionnel mais uniquement un mouvement dans l'image ce qui impacte largement la qualité de la correction adoptée. La correction a en effet tendance bien souvent à surcompenser le mouvement. Cela provoque de nouvelles distorsions qui peuvent aller jusqu'à être aussi flagrantes que les distorsions d'origine. Récemment sont donc apparues des méthodes (Forssén and Ringaby, 2010; Ringaby and Forssén, 2012) utilisant non plus le flot optique mais des points d'intérêts pour estimer le mouvement tridimensionnel global de la scène à chaque instant de l'acquisition de l'un de ces points d'intérêts. Le mouvement est modélisé par une homographie associée à chacun des couples de points d'intérêts. Ces homographies sont ensuite interpolées entre les lignes des points d'intérêts en utilisant une interpolation linéaire de la translation et une interpolation sphérique de la rotation (voir la méthode SLERP présentée annexe C.4).

III.3.1.2 Méthodes utilisant des éléments extérieurs à la caméra

Dans le cas de la robotique, (Nicklin et al., 2007) montre que l'utilisation de capteurs internes au robot peut permettre de calculer directement l'incrément de décalage des lignes horizontales. Lorsque des caméras «Rolling Shutter» sont utilisées pour réaliser simultanément la localisation et la cartographie (SLAM), une estimation du mouvement est disponible en fonction des images précédentes. Dans (Klein and Murray, 2009), cette estimation est utilisée pour calculer un vecteur de translation qui est réestimé dans l'image courante en plus des autres paramètres du SLAM.

En utilisant plusieurs caméras «Rolling Shutter» synchronisées, il est possible de fournir à une cadence élevée des images sans distorsions (Wilburn et al., 2004; Bradley et al., 2009). Pour cela, les lignes qui sont acquises au même moment dans les différentes caméras, toutes décalés du temps d'acquisition d'une ligne, sont assemblées afin de former l'image finale.

L'utilisation de caméras «Rolling Shutter» pour la reconnaissance faciale est aussi problématique, ainsi (Heflin et al., 2010) s'est intéressé à l'impact et la correction de l'effet «Rolling Shutter» dans ce cas précis.

III.3.2 Modélisation de la projection «Rolling Shutter»

Dans le cas d'une caméra «Rolling Shutter», en supposant l'objet filmé connu, il a été montré qu'il était possible d'obtenir à partir d'une seule image aussi bien une estimation de la pose de l'objet que

de son mouvement. Dans le cas d'un mouvement uniforme, ce problème a été résolu par l'optimisation d'une erreur de reprojection non-linéaire (Ait-Aider et al., 2006). Cette méthode a été également appliquée avec succès dans le cadre de l'asservissement visuel. Lorsque l'objet est inconnu et en utilisant le même modèle de mouvement, il a été également montré qu'il était possible de retrouver la structure, la pose et le mouvement d'un objet à partir d'une paire stéréo constituée d'au moins une caméra «Rolling Shutter» (Ait-Aider and Berry, 2009). Dans la prolongée du classique «Structure from Motion», il a aussi été montré qu'il est possible d'étendre l'algorithme de l'ajustement de faisceau («Bundle Adjustement») au cas d'une caméra «Rolling Shutter» permettant ainsi l'obtention à partir de plusieurs images de la structure de la scène, de la pose et du mouvement de la caméra (Hedborg et al., 2012).

III.3.2.1 Estimation de pose en «Rolling Shutter»

Une première réflexion sur l'estimation de pose dans le cas de l'utilisation d'une caméra «Rolling Shutter» est donnée par (Meingast et al., 2005). Celle-ci montre que l'utilisation des algorithmes classiques d'estimation de pose appliqués à des images issues d'une caméra «Rolling Shutter» donne des résultats au mieux de mauvaise qualité, au pire totalement erronés quand ce n'est pas carrément un échec de l'algorithme qui se produit. Les auteurs proposent alors d'ajuster l'estimation de pose en introduisant une modélisation du mouvement dans le modèle de projection. Le mouvement est supposé être fronto-parallèle à la caméra et donc constitué d'une translation parallèle au plan image de vitesse $\begin{bmatrix} v_x & v_y \end{bmatrix}^\top$ et d'une rotation de vitesse ω_z autour de l'axe optique. La matrice de projection P_i à l'instant t_i ainsi obtenue est donnée par

$$\mathsf{P}_i = \mathsf{K} \left[\underbrace{\begin{bmatrix} 1 & -t_i\omega_z & 0 \\ t_i\omega_z & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{=\mathsf{R}_i} \mathsf{R}_0 \quad \mathbf{t}_0 + t_i \begin{bmatrix} v_x \\ v_y \\ 0 \end{bmatrix} \right], \tag{III.1}$$

où K est la matrice d'étalonnage des paramètres internes de la caméra. La matrice R_i n'est pas orthonormale, c'est le développement limité d'une rotation d'angle $t_i\omega_z$, supposé faible, autour de l'axe optique. Cette paramétrisation du mouvement a l'avantage d'être linéaire par rapport au temps, ce qui permet de l'éliminer facilement. Nous obtenons ainsi un terme de correction à appliquer par rapport à la projection «Global Shutter» classique. Celui-ci dépend uniquement de la vitesse d'acquisition des lignes de l'image.

Simultanément, et dans le cas d'un mouvement plus complexe mais néanmoins toujours uniforme, des travaux ont abouti au modèle de projection que nous appellerons «Uniform Rolling Shutter» (Ait-Aider et al., 2006). Ce modèle est plus général et contient plus de degrés de liberté car le mouvement de

rotation de l'objet ou de la scène est paramétré en utilisant la formule de Rodrigues (Rodrigues, 1840) :

$$\delta\mathsf{R}(t, \mathbf{a}, \omega) = \mathbf{a}\mathbf{a}^\top(1 - \cos(t\omega)) + \mathtt{I}\cos(t\omega) + [\mathbf{a}]_\wedge \sin(t\omega), \tag{III.2}$$

où $\delta\mathsf{R}(t, \mathbf{a}, \omega)$ est une matrice qui traduit la rotation à l'instant t autour de l'axe $\mathbf{a}$ à la vitesse ω. L'axe de rotation $\mathbf{a}$ est un vecteur unitaire, c'est-à-dire tel que $\|\mathbf{a}\| = 1$. Le mouvement de translation est paramétré par définition de la vitesse uniforme $\mathbf{v}$. Ainsi la translation à l'instant t est donnée par

$$\delta\mathbf{t}(t, \mathbf{v}) = t\mathbf{v}. \tag{III.3}$$

La projection du point $\mathbf{p}_i$ sur son image $\mathbf{m}_i = \begin{bmatrix} u_i & v_i \end{bmatrix}^\top$ à l'instant $\tau_i = v_i\tau$, où τ est le temps d'exposition des lignes, est donnée par

$$\tilde{\mathbf{m}}_i \sim \mathsf{K}\left[\mathsf{R}_0\delta\mathsf{R}(\tau_i, \mathbf{a}, \omega) \mid \mathbf{t}_0 + \delta\mathbf{t}(\tau_i, \mathbf{v})\right]\tilde{\mathbf{p}}_i, \tag{III.4}$$

où $(\mathsf{R}_0, \mathbf{t}_0)$ est la pose de l'objet ou de la caméra à l'instant $\tau_0 = 0$. Ce modèle de projection est illustré dans la figure III.13.

Afin d'estimer les paramètres de pose et de mouvement, il est proposé de faire une minimisation de l'erreur de reprojection associée à ce modèle en utilisant la méthode de Levenberg-Marquardt (voir section A.3.1). La paramétrisation de la rotation R_0 est réalisée en utilisant un quaternion unitaire (voir annexe C). La contrainte d'unité résultant de cette paramétrisation est ajoutée à l'erreur de reprojection avec un coefficient de pondération empirique. L'initialisation de l'optimisation est obtenue par un calcul de pose standard pour les paramètres de pose. Pour les paramètres de mouvement, une initialisation générique est utilisée : $\omega = 0$, $\mathbf{a} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^\top$ et $\mathbf{v} = \mathbf{0}$.

Dans le cas de la projection de lignes droites, qui deviennent des courbes dans l'image avec l'effet «Rolling Shutter», le même modèle de projection a été étendu (Ait-Aider et al., 2007). Chaque ligne droite tridimensionnelle est modélisée par un point $\mathbf{P}_{k0}$ et un vecteur directeur $\mathbf{L}_k$ de sorte que chaque point de la ligne puisse s'écrire $\mathbf{P}_{ki} = \mathbf{P}_{k0} + \sigma_{ki}\mathbf{L}_k$ où σ_{ki} est un coefficient réel précisant la position du point $\mathbf{P}_{ki}$ sur la ligne. Connaissant les lignes tridimensionnelles et leurs courbes correspondantes dans l'image, il est impossible de déterminer a priori quel est le point de la ligne qui a été projeté sur un point de la courbe, ce qui signifie que les valeurs σ_{ki} sont inconnues. L'erreur de reprojection est alors fonction des paramètres précédents auxquels viennent s'ajouter les σ_{ki}. La minimisation est toujours réalisée selon la même méthode.

Une adaptation du modèle de projection «Uniform Rolling Shutter» a été utilisée dans le cadre de la poursuite visuelle de marqueurs se trouvant sur le bras d'un robot à haute vitesse (Dahmouche et al., 2008, 2009). Pour atteindre des performances temps réel à très haute vitesse, seules des petites régions d'intérêt centrées sur les marqueurs à suivre sont acquises. L'intervalle de temps entre l'acquisition de

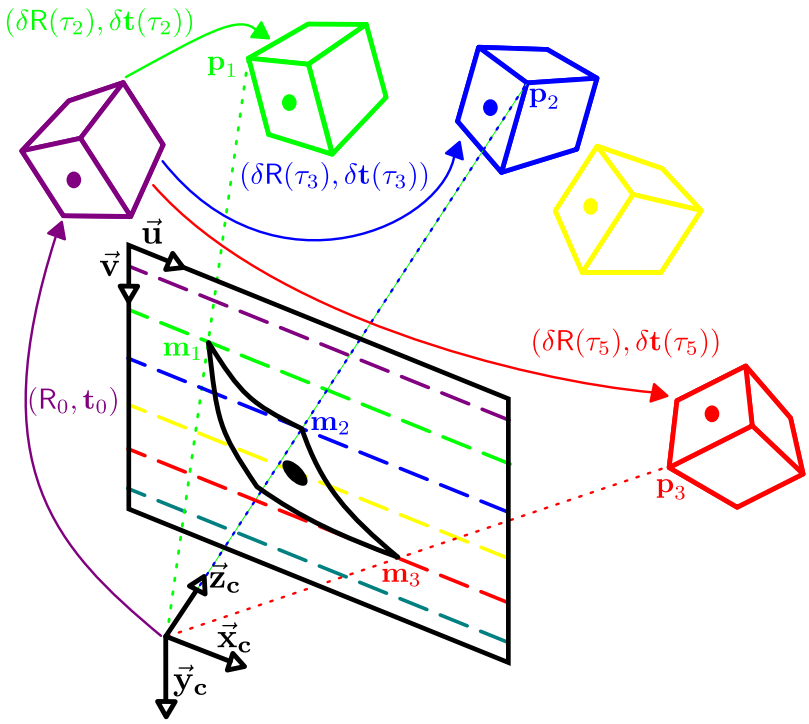


FIGURE III.13 – Illustration de la projection par le modèle «Uniform Rolling Shutter» des points $\mathbf{p}_i$ sur leur image $\mathbf{m}_i$. Pour rendre l'illustration plus lisible, les paramètres du mouvement ont été omis. Ainsi $\delta\mathbf{t}(\tau_i)$ sur la figure est $\delta\mathbf{t}(\tau_i, \mathbf{v})$ et $\delta\mathsf{R}(\tau_i)$ est $\delta\mathsf{R}(\tau_i, \mathbf{a}, \omega)$.

chaque région d'intérêt est connu et considérablement réduit. Leur position est prédite en utilisant le modèle de déplacement uniforme. Une fois les marqueurs détectés sur les différentes régions d'intérêt, une estimation de pose et de mouvement est réalisée, ce qui permet de mettre à jour la prédiction pour les prochaines acquisitions.

III.3.2.2 Estimation de la structure et du mouvement

Les méthodes permettant l'estimation de la structure de l'objet ou de la scène en même temps que l'estimation du mouvement ne sont apparues que très récemment. Tout comme dans le cas «Global Shutter», il est possible de diviser ces méthodes en deux catégories : celles basées sur l'utilisation d'une paire (ou plus) de caméras synchronisées, nous parlons alors de stéréoscopie, et celles basées sur le mouvement d'une seule caméra, nous parlons alors de «Structure-from-Motion».

Dans (Ait-Aider and Berry, 2009) il est présenté une méthode de triangulation pour obtenir à la fois la structure et le déplacement d'un objet dans la scène à partir d'une paire stéréoscopique étalonnée contenant au moins une caméra «Rolling Shutter». L'une des caméras est prise comme référence, ainsi pour chaque ligne la matrice de projection de cette caméra est constituée uniquement du mouvement de l'objet à l'instant de l'acquisition de celle-ci. La matrice de projection de l'autre caméra est constituée de la pose de la deuxième caméra dans le repère de la première, incrémentée du mouvement de l'objet qui peut être calculé à partir du mouvement exprimé dans la caméra de référence. Cela permet d'obtenir une erreur de reprojection pour chaque caméra dépendant de la pose de l'objet, de son mouvement et de sa structure. L'erreur totale est minimisée par une méthode non-linéaire après initialisation par la triangulation «Global Shutter». La mise en correspondance est réalisée manuellement. Il est montré que l'utilisation de deux caméras «Rolling Shutter» crée de nouvelles ambiguïtés par rapport au cas de la triangulation «Global Shutter».

Dans le cas d'une séquence d'images prises par une caméra «Rolling Shutter», il est possible d'adapter le principe du «Structure from Motion» utilisé classiquement avec les caméras «Global Shutter». C'est ce qui est réalisé dans (Hedborg et al., 2011) en considérant que le mouvement de la caméra durant l'acquisition d'une image se limite à une rotation d'angle faible. Cela est justifié par le fait que la rotation induit plus de déformation dans l'image, et donc produit plus d'information. Le mouvement est alors paramétré par une série de rotations clefs à raison d'une au début de chaque image, et les rotations intermédiaires pour chaque lignes sont calculées en utilisant la méthode SLERP (voir annexe C.4). Les rotations clefs sont optimisées en même temps que la pose des caméras pour les différentes images de la séquence. La structure est estimée par triangulation une fois la pose et le mouvement des caméras estimés. Un ajustement de faisceaux (Triggs et al., 2000) est ensuite réalisé pour raffiner l'ensemble. Les correspondances de points sont obtenues à partir de (Lucas and Kanade, 1981) et (Rosten and Drummond, 2006).

L'ajustement de faisceaux peut être adapté à une séquence d'images prises par une caméra «Rolling Shutter» (Hedborg et al., 2012). Cette adaptation s'inspire du même principe que (Hedborg et al., 2011) : à chaque image correspond une pose de référence, celle lors de l'acquisition de la première ligne, et pour chaque ligne d'une image, la pose est calculée en interpolant la pose de référence avec la pose de référence de l'image suivante. Pour la dernière image, il est introduit une pose supplémentaire qui aurait été celle de l'image suivante. Ainsi il n'y a donc que six paramètres supplémentaires par rapport à l'ajustement de faisceaux classique. Les effets sur les matrices Jacobienne et Hessienne de la fonction de coût optimisée sont non-négligeables puisque le modèle introduit une dépendance entre les paramètres de deux images consécutives. Ces matrices restent néanmoins globalement creuses et le calcul des incréments lors de l'optimisation est semblable au cas «Global Shutter».

III.3.2.3 Déformation d'un marqueur

L'utilisation de marqueurs sur le robot est assez classique en asservissement visuel. La question de la précision sur la localisation de ces marqueurs dans l'image est cruciale, puisque d'elle dépend bien souvent en grande partie la qualité de l'estimation de pose, éventuellement dynamique. (Laroche and Kagami, 2009) s'intéresse ainsi à cette problématique. Pour cela ils ont modélisé le processus de déformation du marqueur lors de l'exposition de la rétine à une scène, tant dans le cas du «Global Shutter» que du «Rolling Shutter». Considérant que la position du marqueur durant le temps d'exposition t_e de l'image est paramétrée par $(x(t), y(t))$, la contribution du marqueur à l'image k est donnée par :

- dans le cas «Global Shutter»,

$$\begin{cases} \tilde{x}(k) = \frac{1}{t_e} \int_{t_k - t_e}^{t_k} x(t) \partial t \\ \tilde{y}(k) = \frac{1}{t_e} \int_{t_k - t_e}^{t_k} y(t) \partial t \end{cases}, \tag{III.5}$$

  où t_k est l'instant de fin d'acquisition de l'image ;
- dans le cas «Rolling Shutter»,

$$\begin{cases} \tilde{x}(k) = \frac{1}{t_{r_k}} \int_0^{\inf} C_k(\tau) x(\tau) \partial \tau \\ \tilde{y}(k) = \frac{1}{t_{r_k}} \int_0^{\inf} C_k(\tau) y(\tau) \partial \tau \end{cases}, \tag{III.6}$$

  où t_{r_k} est le temps durant lequel le marqueur apparaît sur l'image k et C_k est une fonction binaire indiquant si le marqueur est présent sur l'image ou non à un instant donné.

Ce modèle a été testé avec un vidéo-projecteur qui diffusait l'image d'un marqueur en rotation autour de l'axe optique du vidéo-projecteur. Le mouvement étant ainsi connu, la mesure de la trajectoire dans l'image du marqueur (qui est alors une portion d'arc-de-cercle) peut être prédite par le modèle et comparée à la vérité terrain ou simulée. Néanmoins aucune application pratique à des problématiques

réelles n'a été considérée.

III.3.3 Problématiques similaires

Il existe d'autres systèmes d'imagerie que les caméras «Rolling Shutter» produisant néanmoins des effets comparables à ceux décrits précédemment. En effet, tout système dans lequel l'acquisition se produit au cours du temps induit une problématique similaire. Nous pouvons citer par exemple l'imagerie satellite qui est réalisée à partir d'une caméra linéaire (un capteur dont la rétine ne contient qu'une seule et unique ligne de sites photosensibles) et qui se déplace avec le satellite. (Hartley and Gupta, 1994) fournit un exemple de modélisation de ce système.

Bien avant le «Rolling Shutter» électronique des caméras CMOS, il a existé des diaphragmes dont l'ouverture de taille fixe est mobile. De telles caméra sont appelées caméras «slit» (l'ouverture est alors une fente qui se déplace selon l'un des axes de l'image) ou «cross-slit» (l'ouverture est alors un trou, souvent formé par l'intersection de deux fentes se déplaçant selon les deux axes). Un modèle de ces dernières est donné par (Pajdla, 2002). Une généralisation de ce modèle qui inclut aussi les caméras linéaires existe (Yu and McMillan, 2004).

Dans le cas de la reconstruction tridimensionnelle de la scène par l'utilisation de capteurs «Time of Flight» ou d'une caméra couplée à de la lumière structurée (cas de la Kinect), l'effet «Rolling shutter» peut être observé. En effet, si le capteur utilise un mode d'acquisition «Rolling Shutter» alors la reconstruction tridimensionnelle d'une scène en mouvement va être perturbée. Des publications récentes (Ringaby and Forssén, 2011) se sont intéressées à la problématique de correction de la reconstruction tridimensionnelle avec de tels capteurs.

CHAPITRE IV

MODÉLISATION «ROLLING SHUTTER» NON-UNIFORME

Résumé

Bien que l'hypothèse d'uniformité du mouvement utilisée dans (Ait-Aider et al., 2006, 2007; Ait-Aider and Berry, 2009) donne de bons résultats, nous pouvons nous interroger sur la pertinence de cette approximation. C'est tout particulièrement vrai en robotique très haute vitesse où les mouvements sont très rapides et où le nombre d'accélérations ou décélérations fortes est très important. Ces accélérations ou décélérations peuvent être très brutales, il n'est donc pas impossible que celles-ci se produisent durant l'acquisition d'une image. La question de l'estimation du mouvement en tenant compte de l'accélération peut alors être intéressante à se poser. Un exemple d'image comportant des mouvements non-uniformes est donné à la figure IV.1.

Dans ce chapitre, nous allons donc nous intéresser à cette problématique dans le cas où un objet est en déplacement rapide devant une caméra «Rolling Shutter». En suivant les lois de la physique, la modélisation complète du mouvement avec accélération nécessite de connaître la matrice d'inertie de l'objet. Ce serait un paramètre de plus à étalonner et qui est propre à chaque objet. Cette matrice ne peut être estimée à partir de l'image en même temps que l'accélération puisque la trajectoire est déterminée par le produit de l'accélération et de la matrice d'inertie. Afin d'éviter de faire un apport d'information a priori, une approche différente est suivie : modéliser le mouvement en paramétrant la pose de l'objet ligne par ligne et créer un lissage des paramètres de pose entre les lignes.

Nous présentons également au cours de ce chapitre un cadre de travail générique qui permet la modélisation de la projection et l'estimation de la pose et du mouvement par optimisation non-linéaire, tant pour les caméras «Global Shutter» que «Rolling Shutter». Ce cadre de travail peut même être utilisé

FIGURE IV.1 – La scène filmée, qui se trouve dans le champ de la caméra, est déformée du probablement à une succession de mouvement non-uniformes de la caméra.

de manière générale avec tout système d'imagerie dont l'exposition ou l'acquisition de tout ou partie de l'image est asynchrone. Nous introduirons pour cela la notion de pose dynamique, où la position de l'objet dans le repère caméra est représenté par une fonction du temps, éventuellement constante, et qu'elle soit continue ou discrète. Nous montrerons comment les travaux précédents peuvent se représenter dans ce cadre de travail, et l'utiliserons également pour présenter notre modélisation «Rolling Shutter» non-uniforme. Ces travaux ont été publiés dans (Magerand and Bartoli, 2010).

IV.1 Calcul de pose dynamique

IV.1.1 Notion de pose dynamique

IV.1.1.1 Définition

Dans la section II.2, la pose a été définie comme étant la position et l'orientation du repère caméra dans le repère monde (ou objet). Ces paramètres peuvent être amenés à changer durant l'acquisition d'une image complète : l'exposition des différents pixels de l'image n'est pas synchrone. Lors de la projection d'un point sur un pixel, il est nécessaire de connaître la position et l'orientation du repère caméra au moment de l'exposition de ce pixel. La notion de pose doit donc être étendue pour prendre en compte cette dépendance au temps.

Nous appellerons *pose dynamique* une pose dont la position et l'orientation sont fonction du temps, à opposer à la notion de pose habituelle que nous appellerons *pose statique*. Cela nécessite généralement de nouveaux paramètres, incluant parfois ceux d'une pose statique, qui définissent les variations de position et d'orientation du repère caméra durant l'acquisition. Notons $\mathbf{x}$ un vecteur contenant l'ensemble des paramètres nécessaires. Il faut noter que selon la paramétrisation de la pose dynamique, la taille de ce vecteur peut être supérieure au nombre de paramètres indépendants de la modélisation du mouvement.

La position et l'orientation ne sont pas nécessairement définies à tout instant et le temps peut être échantillonné. Le domaine de définition de la pose dynamique est un ensemble d'instants de cet échantillonnage. Indexons par $j \in \mathcal{F} \subset \mathbb{Z}$ ces instants qui seront alors notés τ_j. Nous pouvons donc écrire la pose dynamique sous la forme

$$\{(\mathsf{R}(j,\mathbf{x}), \mathbf{t}(j,\mathbf{x}))\}_{j\in\mathcal{F}}\,. \tag{IV.1}$$

IV.1.1.2 Modèle de projection

La projection perspective classique résultant de l'hypothèse trou d'épingle et utilisant une pose statique peut être étendue à l'utilisation de pose dynamique. Néanmoins, afin de pouvoir projeter un point sur un pixel de l'image avec une pose dynamique, il est nécessaire que cette pose dynamique soit définie à l'instant où le pixel est exposé.

En supposant que l'ensemble de définition de la pose dynamique inclut tous les instants $\{j_i\}_{i\in[1;n]}$ où des points se projettent, alors la projection perspective est donnée par

$$\tilde{\mathbf{m}}_i \sim \mathsf{K} \underbrace{[\,\mathsf{R}(j_i,\mathbf{x}) \mid \mathbf{t}(j_i,\mathbf{x})\,]}_{=\mathsf{M}_i(\mathbf{x})} \tilde{\mathbf{p}}_i, \tag{IV.2}$$

où K est la matrice d'étalonnage définie équation (II.9) et j_i l'index de l'instant où $\mathbf{p}_i$ se projette.

L'erreur de reprojection associée à ce modèle de projection est définie de la même manière que pour la projection perspective classique. En notant $\mathbf{q}_i$ les coordonnées de la projection mesurées dans l'image pour le point $\mathbf{p}_i$ et en considérant l'ensemble des points, elle s'exprime par

$$\xi = \sum_{i=1}^{n} \|\mathbf{q}_i - \boldsymbol{\varphi}(\mathsf{M}_i(\mathbf{x})\tilde{\mathbf{p}}_i)\|^2. \tag{IV.3}$$

IV.1.2 Instances de poses dynamiques

IV.1.2.1 Cas d'une pose statique

Dans le cas d'une projection «Global Shutter», il n'existe qu'un seul instant où la pose est définie puisque l'ensemble des pixels de l'image sont exposés au même moment. L'ensemble $\mathcal{F}$ se réduit ainsi au singleton $\{1\}$. Le vecteur paramètres $\mathbf{x}$ contient quand à lui les paramètres de la pose $(\mathsf{R}, \mathbf{t})$. La pose dynamique s'écrit alors

$$\begin{cases} \mathsf{R}(1, \mathbf{x}) = \mathsf{R} \\ \mathbf{t}(1, \mathbf{x}) = \mathbf{t}. \end{cases} \tag{IV.4}$$

IV.1.2.2 Cas d'un mouvement uniforme

Lorsque le mouvement du repère caméra dans le repère monde est uniforme, (Ait-Aider et al., 2006; Ait-Aider and Berry, 2009) fournit une paramétrisation du mouvement qui peut se mettre sous forme d'une pose dynamique selon

$$\begin{cases} \mathsf{R}(j, \mathbf{x}) = \mathsf{R}_0 \delta\mathsf{R}(\tau_j, \mathbf{x}') \\ \mathbf{t}(j, \mathbf{x}) = \mathbf{t}_0 + \delta\mathbf{t}(\tau_j, \mathbf{x}'), \end{cases} \tag{IV.5}$$

où $\mathbf{x}$ contient une vectorisation de la pose initiale $(\mathsf{R}_0, \mathbf{t}_0)$ ainsi que $\mathbf{x}'$, les sept paramètres de vitesse ($\mathbf{v}, \mathbf{a}, \omega$ de la section III.3.2.1). Cette pose dynamique est définie pour l'ensemble des lignes de l'image et donc $\mathcal{F} = [1; l]$, l étant le nombre de lignes de l'image.

Une autre paramétrisation possible pour le mouvement uniforme est proposée par (Hedborg et al., 2011, 2012). Dans ce cas, le vecteur $\mathbf{x}$ contient les paramètres des poses $(\mathsf{R}_0, \mathbf{t}_0)$ et $(\mathsf{R}_l, \mathbf{t}_l)$. La première correspond à la pose lors de l'exposition de la première ligne de l'image, et la dernière lors de l'exposition

de la dernière ligne de l'image (ligne l) [1]. La pose dynamique correspondante est alors donnée par

$$\begin{cases} \mathsf{R}(j, \mathbf{x}) = \mathrm{slerp}(\mathsf{R}_0, \mathsf{R}_l, j/l) \\ \mathbf{t}(j, \mathbf{x}) = \mathbf{t}_0 + \frac{j}{l}\mathbf{t}_l, \end{cases} \tag{IV.6}$$

où $\mathrm{slerp}()$ désigne l'interpolation sphérique linéaire décrite dans l'annexe C.4.

IV.1.3 Contraintes et optimisation

IV.1.3.1 Origine des contraintes

L'erreur de reprojection fournit deux contraintes par point considéré. Pour que le problème de l'estimation du vecteur $\mathbf{x}$ à partir de cette erreur soit bien posé, il faut donc que la taille de ce dernier soit inférieure à $2n$, n étant le nombre de points utilisés. Dans le cas contraire il va être nécessaire d'ajouter au problème des contraintes. Ces dernières découlent bien souvent de réalités physiques qui s'imposent aux objets. Un exemple est donné par l'impossibilité pour un objet solide de se déformer.

Lorsque la taille du vecteur $\mathbf{x}$ est supérieure au nombre de paramètres indépendants du modèle de mouvement, alors il existe aussi des contraintes liant les paramètres supplémentaires. Dans ce cas il est également nécessaire d'imposer ces contraintes lors de l'optimisation de l'erreur de reprojection. Celles-ci proviennent soit de limites physiques, soit de propriétés mathématiques. Typiquement, la paramétrisation de l'orientation par une rotation nécessite généralement d'imposer que la matrice de taille 3×3 qui la représente soit bien orthonormale.

L'ensemble de toutes ces contraintes sera noté par la suite sous forme d'un vecteur $\mathbf{C}(\mathbf{x})$ qui doit être nul.

IV.1.3.2 Méthode d'optimisation

En conservant les notations des sections précédentes, le problème à résoudre est donné par l'optimisation

$$\min_{\mathbf{x}} \sum_{i=1}^{n} \| \mathbf{q}_i - \boldsymbol{\varphi}\left(\mathsf{M}_i(\mathbf{x})\tilde{\mathbf{p}}_i\right) \|^2, \quad \textit{s.c.} \quad \mathbf{C}(\mathbf{x}) = 0. \tag{IV.7}$$

En l'absence de contraintes, cette optimisation non-linéaire est un cas typique où l'utilisation de la méthode de Levenberg-Marquardt présentée section A.3.1 est parfaitement adaptée. Dans le cas contraire, si nous souhaitons conserver la méthode de Levenberg-Marquardt, il est nécessaire de former une relaxation du problème en utilisant une fonction de coût augmentée de pénalités, qui sont les contraintes.

1. Travaillant sur une séquence vidéo, les auteurs de (Hedborg et al., 2011, 2012) ont en fait considéré les poses à l'instant de l'acquisition de la première ligne de l'image courante et de la première ligne de l'image suivante.

Le problème à résoudre devient alors

$$\min_{\mathbf{x}}(1-\lambda)\sum_{i=1}^{n}\|\mathbf{q}_i-\boldsymbol{\varphi}\left(\mathsf{M}_i(\mathbf{x})\tilde{\mathbf{p}}_i\right)\|^2+\lambda\|\mathbf{C}(\mathbf{x})\|^2, \tag{IV.8}$$

où $\lambda \in]0;1[$ est un coefficient de régularisation permettant d'ajuster l'effet des pénalités par rapport à la fonction de coût initiale. Une valeur de λ proche de 1 privilégie les pénalités par rapport à la recherche de la valeur minimisant la fonction de coût originale tandis qu'une valeur faible aura l'effet inverse.

En présence de contraintes, une autre solution est de changer de méthode d'optimisation pour une méthode intégrant la gestion des contraintes. La méthode par programmation quadratique séquentielle permet de résoudre de tels problèmes. Comme expliqué dans la section A.3.2, la méthode par programmation quadratique séquentielle forme une fonction de coût augmentée utilisant les multiplicateurs de Lagrange. La différence réside principalement dans le fait que cette méthode transforme les multiplicateurs de Lagrange en des inconnues du problème qui sont estimées et les contraintes sont ainsi parfaitement satisfaites.

IV.1.4 Pose dynamique non-uniforme

À partir du moment où le temps est correctement échantillonné, il est possible de représenter n'importe quel mouvement avec la pose dynamique la plus générique possible : celle qui fait correspondre une pose statique à chaque instant. Dans ce cas, le vecteur $\mathbf{x}$ contient les paramètres des différentes poses, c'est-à-dire six paramètres indépendants pour chaque instant considéré. Pour le «Rolling Shutter», l'ensemble $\mathcal{F} = [1;l]$ où l est le nombre de lignes de l'image. La pose dynamique s'écrit alors simplement

$$\{(\mathsf{R}_j,\mathbf{t}_j)\}_{j=1}^{l}. \tag{IV.9}$$

Étant donné le nombre de paramètres, pour estimer une telle pose dynamique, il ne faudrait donc pas moins de trois points projetés sur chaque ligne. Ces points seraient de toute manière colinéaires et ne suffiraient donc pas à estimer une pose par ligne. Afin de résoudre ce problème et de réduire le nombre de points nécessaires, il est possible d'imposer des contraintes sur l'évolution des paramètres de pose au court du temps. En effet un objet, ou la caméra, ne peuvent se déplacer dans la scène qu'en suivant une trajectoire continue. Il est donc raisonnable de supposer que la dérivée de l'évolution des paramètres au court du temps est nulle à un certain degré d. En utilisant une dérivation centrale finie

pour les instants $j \in [1 + \lceil d/2 \rceil, l - \lfloor d/2 \rfloor]$, cela s'écrit alors

$$\mathbf{C}(\mathbf{x}) = \begin{bmatrix} \delta^d \mathbf{x}_{1+\lceil d/2 \rceil} \\ \vdots \\ \delta^d \mathbf{x}_{l-\lfloor d/2 \rfloor} \end{bmatrix} = \mathbf{0}, \tag{IV.10}$$

$$\delta^d \mathbf{x}_j = \sum_{h=0}^{d} (-1)^h \binom{d}{h} \mathbf{x}_{j+\lfloor \frac{d}{2} \rfloor - h}, \tag{IV.11}$$

où $\mathbf{x}_j$ est la partie du vecteur $\mathbf{x}$ contenant les paramètres de $(\mathsf{R}_j, \mathbf{t}_j)$.

Pour donner un sens plus précis à cet a priori, il faut s'intéresser à la paramétrisation des poses $(\mathsf{R}_j, \mathbf{t}_j)$. Pour les translations, nous avons utilisé classiquement des vecteurs indiquant la position du centre du repère objet dans le repère caméra. Réaliser leurs différences finies centrées par rapport aux instants revient donc à faire une dérivation à l'ordre d de cette position. L'a priori correspond ainsi à imposer que le mouvement du centre du repère objet dans le repère caméra se fasse avec une dérivée d-ième nulle. Pour $d = 1$, il n'y a donc pas de mouvement de ce centre. Le cas $d = 2$ nous donne un mouvement à vitesse constante tandis que $d = 3$ nous donne un mouvement à accélération constante et enfin $d = 4$ implique un mouvement à jerk constant, ce qui est une contrainte fréquemment rencontrée en robotique très haute vitesse.

En ce qui concerne les rotations, nous avons utilisé des quaternions sous forme de vecteurs à quatre coordonnées comme décrit dans l'annexe C.3. De par les propriétés de ces derniers, le premier coefficient correspond au cosinus de la moitié de l'angle θ de la rotation. Pour $d = 1$, nous imposons ainsi que

$$\frac{d \cos(\frac{\theta}{2})}{dt} = -\frac{d\theta}{dt} \frac{\sin(\frac{\theta}{2})}{2} = 0, \tag{IV.12}$$

ce qui revient donc à demander que soit $\frac{d\theta}{dt} = 0$ et donc la rotation se fasse à angle constant, soit $\sin(\frac{\theta}{2}) = 0$ et la rotation est nulle. Dans le cas $d = 2$, nous obtenons

$$\frac{\partial^2 \theta}{\partial t^2} \sin(\theta) + \frac{d\theta}{dt}^2 \cos(\theta) = 0. \tag{IV.13}$$

Cette condition est vérifiée de toute évidence lorsque $\frac{d\theta}{dt} = 0$ comme précédemment car $\frac{d\theta}{dt} = 0 \Rightarrow \frac{\partial^2 \theta}{\partial t^2} = 0$. Il n'y a pas d'autre interprétation simple puisque θ est solution d'une équation différentielle du second degré qui est non-linéaire.

Pour l'axe de la rotation, qui est donné par le vecteur formé des trois derniers coefficients du quaternion, nous obtenons un axe fixe lorsque $d = 1$. Dans le cas $d = 2$, l'axe se déplacera à vitesse constante alors que dans le cas $d = 3$ il évoluera avec une accélération constante.

IV.2 Détails de l'implémentation

IV.2.1 Coefficient des contraintes

Lorsque nous utilisons la relaxation décrite équation (VI.2) avec une fonction de coût augmentée des contraintes, il se pose le problème de la détermination du coefficient de régularisation. C'est un problème qui a été beaucoup étudié, et trois grandes catégories de méthodes permettent de le faire : le principe de Morozov (Bonesky, 2009; Lu et al., 2010), les multiples variantes de la validation croisée (Wahba and Wold, 1975; Kohavi, 1995) et enfin les méthodes basées sur l'étude des «L-Curves» (Hansen, 1992; Vogel, 1996; Hansen, 2005).

Ces trois méthodes souffrent d'un inconvénient majeur : elles sont extrêmement lourdes à mettre en oeuvre bien que très simples à implémenter. Des progrès ont cependant été réalisés dans ces dernières années, et des méthodes approximant les résultats de la validation croisée ou des «L-Curves» avec des performances satisfaisantes sont apparues. L'une d'elle, la «L-Tangent Norm» (Brunet et al., 2008), basée sur les «L-Curves» a retenu notre attention.

Rappelons tout d'abord que le principe de la méthode des «L-Curves» est de trouver le meilleur compromis possible entre l'optimisation de la fonction de coût initiale, appelée terme de données, et la satisfaction des contraintes, aussi appelé terme de lissage. Pour cela nous étudions l'évolution de la courbe paramétrique définie par le logarithme de la fonction de coût en fonction du coefficient de régularisation et celui de l'erreur sur les contraintes en fonction de ce même coefficient. Comme nous pouvons le voir dans la figure IV.2, dans des conditions idéales la courbe résultante du tracé de cette évolution ressemble normalement vaguement à la lettre «L» pour un bruit faible, d'où le nom de la méthode.

Notons $\rho(\lambda)$ le logarithme de la fonction de coût en fonction de λ, le coefficient de régularisation, et $\eta(\lambda)$ celui de l'erreur sur les contraintes. La «L-Curve» est donc la courbe paramétrique définie par

$$\begin{bmatrix} \rho(\lambda) & \eta(\lambda) \end{bmatrix}^\top \in \mathbb{R}_+^2, \lambda \in]0;1[. \tag{IV.14}$$

La valeur de λ recherchée est celle qui correspond au coin de la lettre «L» et qui donc maximise la courbure (Spivak, 1999) de la «L-Curve» qui est définie par

$$\kappa(\lambda) = 2\frac{\frac{d\rho(\lambda)}{d\lambda}\frac{\partial^2\eta(\lambda)}{\partial\lambda^2} - \frac{\partial^2\rho(\lambda)}{\partial\lambda^2}\frac{d\eta(\lambda)}{d\lambda}}{\left(\frac{d\rho(\lambda)}{d\lambda}^2 + \frac{d\eta(\lambda)}{d\lambda}^2\right)^{3/2}}. \tag{IV.15}$$

Dans le cas parfait, cette fonction n'admet qu'un seul maximum. Cependant en présence de bruit ou d'autres éléments perturbants, il arrive qu'il y ait plusieurs maxima. Le choix du bon maximum, et donc

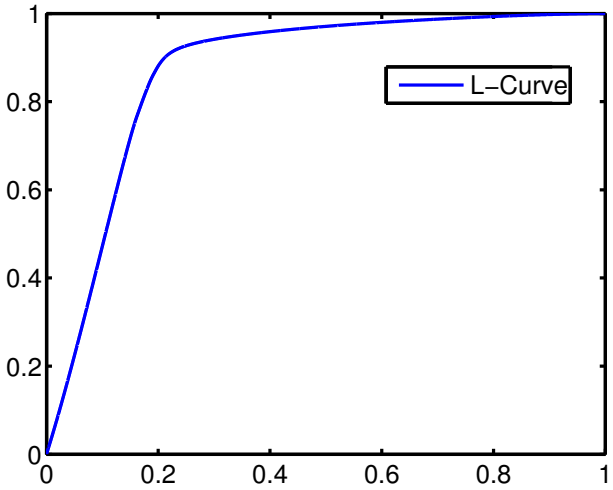


FIGURE IV.2 – Exemple du tracé de la fonction de coût en fonction du coefficient de régularisation. Ici les conditions sont idéales et la courbe ressemble à la lettre «L» renversée.

du bon coefficient de régularisation, n'est alors pas évident et aucune condition n'existe pour ce faire.

Le principe de la «L-Tangent Norm» est de rechercher non plus l'endroit de la «L-Curve» tel que la courbure soit maximum, mais plutôt celui où l'impact d'une variation du coefficient de régularisation sur la variation de la «L-Curve» est minimal. Autrement dit, c'est l'endroit où le compromis entre la fonction de coût et les contraintes est le plus stable. Nous recherchons donc une valeur de λ telle que la norme du vecteur tangent à la courbe soit minimale. Cette norme est déterminée par

$$L(\lambda) = \left\| \begin{bmatrix} \frac{d\hat{\rho}(\lambda)}{d\lambda} & \frac{d\hat{\eta}(\lambda)}{d\lambda} \end{bmatrix}^\top \right\|_2^2 \tag{IV.16}$$

où $\hat{\rho}(\lambda)$ et $\hat{\eta}(\lambda)$ sont les normalisations de $\rho(\lambda)$ et $\eta(\lambda)$ données par

$$\hat{\rho}(\lambda) = \frac{\rho(\lambda) - \rho(\varepsilon)}{\rho(1-\varepsilon) - \rho(\varepsilon)} \quad \hat{\rho}(\lambda) = \frac{\rho(\lambda) - \rho(\varepsilon)}{\rho(1-\varepsilon) - \rho(\varepsilon)}, \tag{IV.17}$$

avec ε une constante positive de faible valeur. Cette normalisation est nécessaire pour éviter les problèmes d'échelle entre la fonction de coût et l'erreur de contrainte. Un exemple de courbe de la norme L-Tangente se trouve figure IV.3.

IV.2.2 Calcul des matrices Jacobiennes

Pour des raisons d'efficacité, il est préférable de calculer analytiquement la matrice Jacobienne de la fonction optimisée. Cette dernière étant composée de plusieurs termes, la matrice Jacobienne peut se diviser en plusieurs sous-matrices empilées verticalement : dérivées de l'erreur de reprojection, dérivées

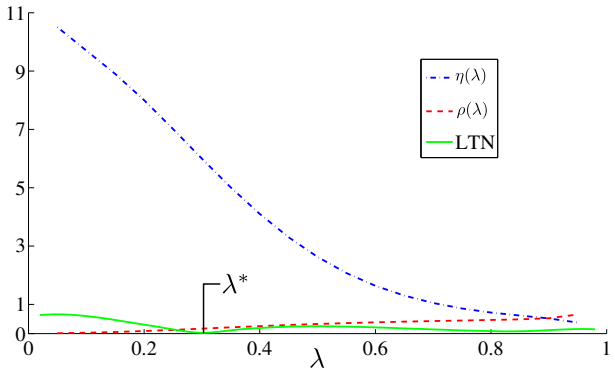


FIGURE IV.3 – Exemple du tracé de la norme L-Tangente en fonction du coefficient de régularisation. La valeur recherchée est telle que la norme de la tangente à la courbe paramétrique $\left[\begin{array}{cc} \frac{d\hat{\rho}(\lambda)}{d\lambda} & \frac{d\hat{\eta}(\lambda)}{d\lambda} \end{array} \right]^\top$ soit minimale.

des contraintes de lissage et dérivées des contraintes d'unité des quaternions. Un exemple de la matrice jacobienne finale est présenté figure IV.4. Ces sous-matrices sont éparses, et en ordonnant correctement le vecteur $\mathbf{x}$, elles deviennent diagonale par bande. Cela permet d'accélérer certains calculs durant l'optimisation.

En ce qui concerne l'erreur de reprojection définie équation (IV.3) et en reprenant les notations de la section II.2, nous pouvons la réécrire pour un point donné sous la forme

$$\left\{ \begin{array}{l} \xi_i^u = u_i - \alpha_u \frac{\mathbf{r}_{v_i}^1 \mathbf{p}_i + t_{v_i}^1}{\mathbf{r}_{v_i}^3 \mathbf{p}_i + t_{v_i}^3} + u_0 \\ \xi_i^v = v_i - \alpha_v \frac{\mathbf{r}_{v_i}^2 \mathbf{p}_i + t_{v_i}^2}{\mathbf{r}_{v_i}^3 \mathbf{p}_i + t_{v_i}^3} + v_0, \end{array} \right. \tag{IV.18}$$

avec $\mathbf{q}_i = [u_i, v_i]^\top$ et où $\mathbf{r}_j^k$ désigne la ligne k de la matrice de rotation à l'instant j, et t_j^k l'entrée k de la translation à ce même instant. Les dérivées partielles de l'erreur de reprojection par rapports aux paramètres de translation sont alors données par

$$\left\{ \begin{array}{ll} \frac{d\xi_i^u}{dt_{v_i}^1} = \frac{-\alpha_u}{\mathbf{r}_{v_i}^3 \mathbf{p}_i + t_{v_i}^3} & \\ \frac{d\xi_i^u}{dt_{v_i}^2} = 0 & \\ \frac{d\xi_i^u}{dt_{v_i}^3} = \alpha_u \frac{\mathbf{r}_{v_i}^1 \mathbf{p}_i + t_{v_i}^1}{(\mathbf{r}_{v_i}^3 \mathbf{p}_i + t_{v_i}^3)^2} & \\ \frac{d\xi_i^v}{dt_{v_i}^1} = 0 & \\ \frac{d\xi_i^v}{dt_{v_i}^2} = \frac{-\alpha_v}{\mathbf{r}_{v_i}^3 \mathbf{p}_i + t_{v_i}^3} & \\ \frac{d\xi_i^v}{dt_{v_i}^3} = \alpha_v \frac{\mathbf{r}_{v_i}^2 \mathbf{p}_i + t_{v_i}^2}{(\mathbf{r}_{v_i}^3 \mathbf{p}_i + t_{v_i}^3)^2}. & \end{array} \right. \tag{IV.19}$$

Tandis que les dérivées partielles par rapports aux paramètres de rotation sont toutes identiques quelque

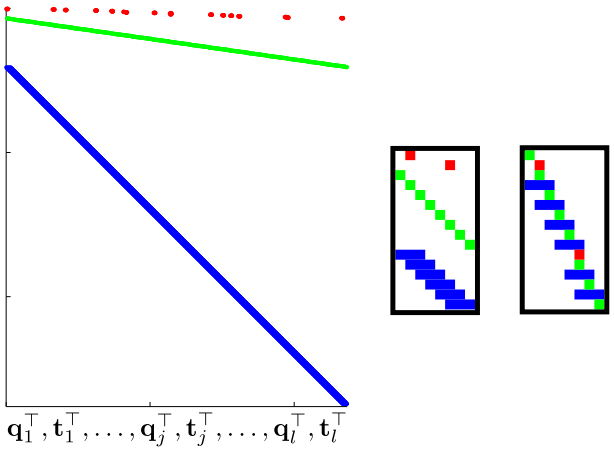


FIGURE IV.4 – Exemple de structure de la matrice Jacobienne de la fonction de coût optimisée avec la méthode de Levenberg-Marquardt. En abscisse se trouvent les paramètres des poses pour les différentes lignes, et en ordonnée les différentes dérivées réparties en trois blocs. Ici nous avons d'abord celles correspondant à l'erreur de reprojection (en rouge), suivie des contraintes d'unité des quaternions (en vert), et enfin celles des contraintes de lissage (en bleu). Cette matrice est éparse, et en réorganisant correctement ses lignes, nous pouvons obtenir une matrice de forme diagonale par bande comme sur l'illustration de droite.

soit la paramétrisation adoptée pour les rotations :

$$\begin{cases} \frac{d\xi_i^u}{d\chi} = -\alpha_u \frac{\frac{d\mathbf{r}_{v_i}^1}{d\chi}\mathbf{p}_i(\mathbf{r}_{v_i}^3\mathbf{p}_i+t_{v_i}^3)-(\mathbf{r}_{v_i}^1\mathbf{p}_i+t_{v_i}^1)\frac{d\mathbf{r}_{v_i}^3}{d\chi}\mathbf{p}_i}{(\mathbf{r}_{v_i}^3\mathbf{p}_i+t_{v_i}^3)^2} \\ \frac{d\xi_i^v}{d\chi} = -\alpha_v \frac{\frac{d\mathbf{r}_{v_i}^2}{d\chi}\mathbf{p}_i(\mathbf{r}_{v_i}^3\mathbf{p}_i+t_{v_i}^3)-(\mathbf{r}_{v_i}^2\mathbf{p}_i+t_{v_i}^2)\frac{d\mathbf{r}_{v_i}^3}{d\chi}\mathbf{p}_i}{(\mathbf{r}_{v_i}^3\mathbf{p}_i+t_{v_i}^3)^2}, \end{cases} \tag{IV.20}$$

où χ est le paramètre par rapport auquel nous dérivons. Selon la paramétrisation choisie pour les rotations, les dérivées de la matrice de rotation par rapport à chaque paramètre sont données dans l'annexe C.

Dans la définition des contraintes équation (IV.11), les paramètres du modèle n'apparaissent qu'au premier degré. Une fois les contraintes dérivées, la sous-matrice qui correspond aux contraintes est donc constante et de forme diagonale par bande. Les entrées de cette sous-matrice dépendent uniquement des valeurs de d et l. Par exemple pour $d = 2$ et $l = 5$, nous obtenons pour les paramètres de translation la sous-matrice

$$\begin{bmatrix} 1 & -2 & 1 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & 1 & -2 & 1 \end{bmatrix}. \tag{IV.21}$$

Dans la figure IV.4, cette sous-matrice correspondrait au bloc bleu.

Les contraintes d'unité des quaternions des rotations sont facilement dérivées. La sous-matrice cor-

respondante dépend bien évidemment uniquement des quaternions. Pour chaque quaternion, la dérivée par rapport à l'un de ses paramètres est égale à deux fois ce paramètre. Nous obtenons ainsi pour chaque quaternion une matrice 4×4 diagonale dont les valeurs sont celles du quaternion multipliées par deux.

IV.3 Initialisation «Global Shutter» ou à uniformité interpolée par morceaux

Comme toute optimisation par méthode itérative, la méthode présentée dans ce chapitre nécessite le calcul d'une solution initiale qui sera ensuite raffinée. La solution finale sera atteinte d'autant plus vite et plus sûrement que cette solution initiale est proche de la valeur recherchée. L'initialisation pourrait être réalisée en utilisant le modèle «Rolling Shutter» uniforme. Néanmoins pour des raisons d'efficacité, le calcul de la solution initiale doit être le plus rapide possible, idéalement en temps linéaire. Nous décrivons ici deux méthodes d'initialisation qui satisfont cet impératif, la première linéaire par rapport au nombre de points en correspondances, et la deuxième quasi-linéaire par rapport au nombre de lignes de l'image.

IV.3.1 Principe

L'intuition physique veut que plus l'intervalle de temps sur lequel nous considérons un mouvement est réduit, et moins celui-ci est perceptible. Lorsque l'intervalle de temps tend vers zéro, le mouvement s'efface et l'objet est statique. Expérimentalement, il est possible de remarquer que l'estimation de pose statique donne un résultat d'autant plus proche de la pose moyenne d'un objet en mouvement que l'intervalle de temps écoulé entre l'exposition des différents points utilisés pour l'estimation est faible. Ainsi l'estimation de pose statique en utilisant des points répartis sur toute une image «Rolling Shutter» est peu proche de la pose moyenne de l'objet sur toute la durée de l'exposition de l'image, tandis que l'estimation en utilisant des points répartis sur quelques lignes est plus proche de la pose moyenne de l'objet durant l'acquisition de ces quelques lignes. Une expérience simple et rapide permet de s'assurer de ce fait, le résultat est présenté dans la figure IV.5.

En se fondant sur ces remarques, il est ainsi tout naturel de diviser l'image en plusieurs bandes de quelques lignes, chacune contenant un nombre minimum de points qui serviront à évaluer une pose statique moyenne pour cette bande que nous attribuerons à une ligne judicieusement choisie. Ce principe est illustré dans la figure IV.6.

La division de l'image en bandes peut être réalisée de plusieurs manières, mais doit satisfaire la contrainte suivante : dans chaque bande il faut assez de points pour réaliser l'estimation de pose statique, soit quatre. De plus, il va de soi que plus les bandes sont petites et recouvrent un minimum de ligne, meilleure sera l'estimation finale du point de vue de l'hypothèse réalisée, bien que l'estimation puisse être plus difficile. Des méthodes modernes telles que EPnP (Lepetit et al., 2009) sont cependant de moins en moins perturbées par cet aspect. La méthode que nous avons retenue pour créer ces bandes est la plus simple à implémenter : les points étant classés par la ligne sur laquelle ils sont projetés,

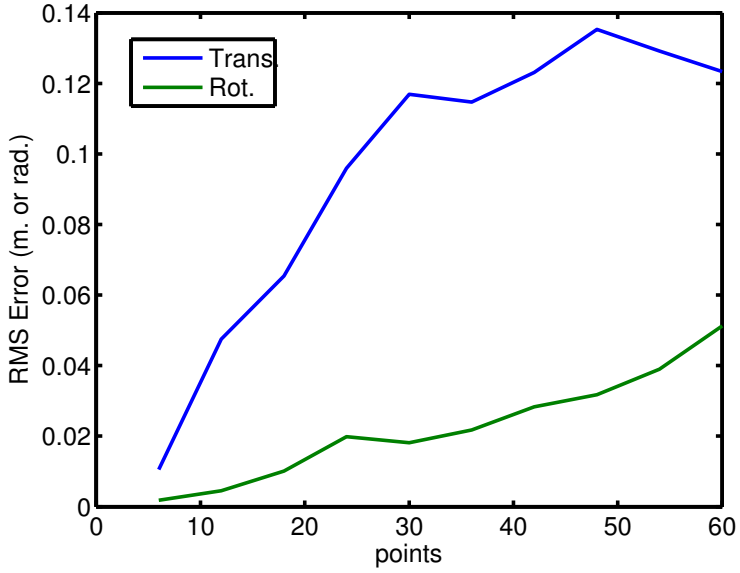


FIGURE IV.5 – La validité de l'hypothèse «Global Shutter» par morceaux peut être étudiée par une simple expérience. Considérons un jeu d'images «Rolling Shutter» simulées, obtenues comme décrit dans le chapitre VI, et définissons pour chacune l'instant moyen auquel les points de l'objet sont exposés. La pose à cet instant est alors comparée à celle obtenue par une estimation «Global Shutter» réalisée en utilisant les n points dont l'instant d'exposition est le plus proche de cet instant moyen. Nous pouvons alors remarquer que la courbe moyenne de l'erreur RMS sur les paramètres de translation ou de rotation augmente à mesure que nous utilisons des points de plus en plus éloignés de la pose moyenne.

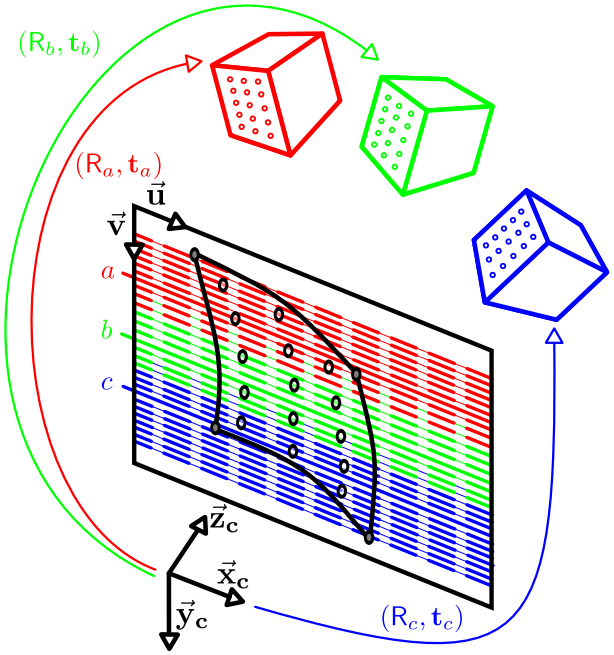


FIGURE IV.6 – Principe de la méthode d'initialisation «Global Shutter» par morceaux : l'image est divisée en différentes bandes (ici rouge, verte et bleue) pour lesquelles une estimation de pose statique est réalisée avec les points projetés dans chaque bande. La pose statique résultante est assignée à la ligne moyenne des lignes où sont projetés les points (ici les lignes a, b, c).

chaque groupe de $n \geq 6$ points consécutifs délimite une bande. Nous entendons ici par points consécutifs des points qui se suivent si l'ensemble des points est trié par leur ligne de projection. La ligne de la pose moyenne est choisie comme étant la moyenne des lignes où se projettent les points. L'algorithme complet de la méthode implémentée est présenté dans la figure IV.7.

1 Séparer les n points en correspondance en $\lfloor n/s \rfloor$ ensembles de s points consécutifs non-recouvrants ;
2 $\mathcal{M} \leftarrow \varnothing$; `// Liste des lignes moyennes de chaque ensemble`
3 **pour chaque** *Ensemble de points* **faire**
4 Calculer j, la ligne moyenne de l'ensemble ;
5 $\mathcal{M} \leftarrow \mathcal{M} \cup \{j\}$;
6 Estimer la pose moyenne $(\mathsf{R}_j, \mathbf{t}_j)$ en utilisant EPnP (Lepetit et al., 2009) avec les s points de l'ensemble ;

FIGURE IV.7 – Algorithme d'estimation de la pose dynamique par la méthode «Global Shutter» par morceaux.

IV.3.2 Validité

Comme indiqué dans la section précédente, plus les points utilisés pour l'estimation de la pose statique se trouvent sur des lignes proches, et plus la pose estimée est proche de la pose moyenne réelle sur cette bande. Cela nous amène à nous intéresser au domaine de validité de la méthode présentée précédemment. Deux questions se posent : tout d'abord quel est le nombre idéal de points dans chaque bande pour que l'estimation soit de qualité suffisante, et ensuite quelle est la taille maximale admissible pour chaque bande, ce qui revient à s'intéresser à la répartition des points sur l'image.

Deux éléments nous permettent de déterminer le nombre idéal de points dans chaque bande. En premier lieu l'estimation étant basée sur l'utilisation de EPnP, l'article de référence sur cette méthode (Lepetit et al., 2009) nous indique que l'estimation est de qualité satisfaisante (moins de 10% d'erreur) à partir de 7 points utilisés pour l'estimation et qu'au delà de 20 points, le gain en précision devient négligeable. Ensuite une rapide expérience a été réalisée sur une trentaine d'images «Rolling Shutter» simulées. Pour chaque image la valeur de s telle que l'erreur relative à la pose réelle soit minimale est recherchée. Les résultats de cette expérience sont présentés dans la figure IV.8, ce qui nous permet de voir que l'intervalle $[8, 18]$ donne habituellement les meilleurs résultats.

Bien que le nombre total de correspondances de points ait son importance, il y a un paramètre qui joue davantage sur la qualité du résultat, c'est leur répartition le long des lignes de l'image. D'après les remarques précédentes, il est naturel de constater que plus les points sont éloignés les uns des autres, et moins l'estimation sera de qualité satisfaisante. Par conséquent il est préférable que la densité de points

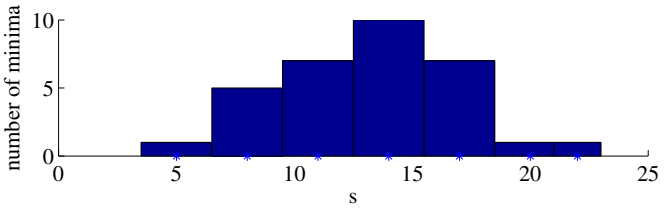


FIGURE IV.8 – En utilisant le jeu de données simulées présenté section VI.1.3.1, nous avons recherché pour chaque image la valeur de s conduisant à l'erreur minimale par rapport à la vérité. L'histogramme montre la distribution du nombre d'images pour chaque valeur de s menant à l'erreur minimale. Nous pouvons constater que habituellement la meilleure valeur de s est comprise dans l'intervalle $[7, 18]$.

dans l'image le long des lignes soit élevée. Il est très aisé de constater l'effet de cette densité de points sur la figure IV.9 qui présente les courbes d'erreurs sur les paramètres estimés en fonction des lignes. Un histogramme de la distribution des points mis en vis-à-vis permet immédiatement de se rendre compte de l'importance de la densité, et ce même au sein d'une image. Plus de résultats seront présentés dans le chapitre VI.

IV.3.3 Complexité

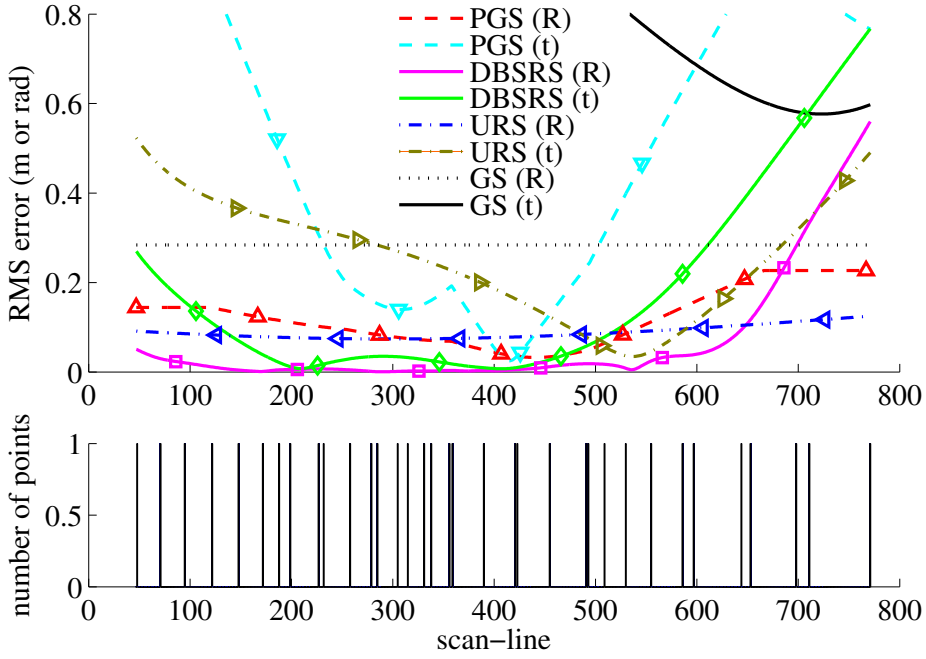
Le calcul de la complexité de cette méthode d'initialisation est aisé. Nous réalisons $\lfloor n/s \rfloor$ estimations de pose avec EPnP, utilisant chacune s points. La complexité de la méthode EPnP est linéaire, autrement dit, nous réalisons $\lfloor n/s \rfloor$ fois une opération en $\mathcal{O}(s)$. La complexité résultante est donc

$$\mathcal{O}(\lfloor n/s \rfloor s) \approx \mathcal{O}(n). \tag{IV.22}$$

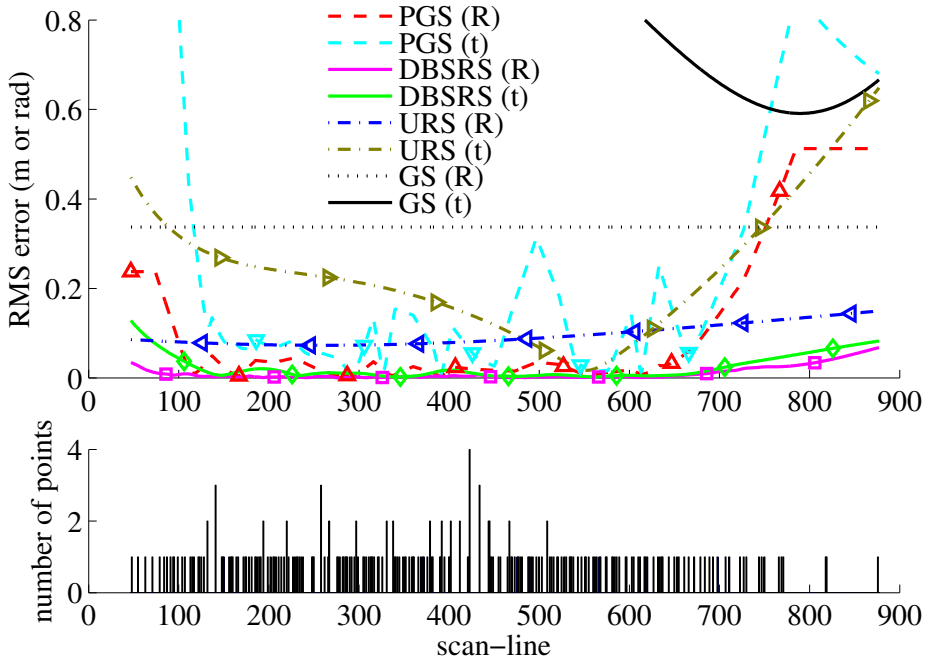
La méthode d'initialisation «Global Shutter» par morceaux est donc de complexité linéaire par rapport au nombre total de points dans l'image.

IV.3.4 Modèle à uniformité interpolée par morceaux

Il est possible de raffiner le modèle d'initialisation présenté précédemment pour le rendre uniforme par morceaux, l'uniformité étant obtenue par interpolation des poses «Global Shutter» entre les lignes des poses moyennes. Nous procédons pour cela à une interpolation linéaire des paramètres de translation entre deux lignes consécutives j_k et j_{k+1} de $\mathcal{M}$, et à une interpolation sphérique en utilisant une interpolation sphérique linéaire (SLERP, voir annexe C.4). La méthode complète est illustrée dans la figure IV.10 et l'algorithme correspondant donné dans la figure IV.11. À la complexité $\mathcal{O}(n)$ de la



(a) Pour une distribution éparse des points mis en correspondance sur l'image, l'estimation «Global Shutter» par morceaux est sujette à de larges erreurs.



(b) Pour le même mouvement et une distribution dense des points mis en correspondance, l'estimation «Global Shutter» par morceaux donne de bons résultats.

FIGURE IV.9 – Les histogrammes illustrent la répartition des points mis en correspondance sur l'image. Les graphes montrent l'erreur quadratique moyenne par rapport à la vérité terrain. La méthode «Global Shutter» par morceaux (PGS) est très sensible à la répartition des points sur l'image, y compris au sein même d'une image. Avec une distribution dense des points, cette méthode fournit une bonne initialisation pour la méthode d'estimation de pose dynamique générique présentée précédemment (DBSRS). Pour référence, les méthodes «Global Shutter» (GS) et le modèle «Rolling Shutter» uniforme (URS) sont également représentées. Dans cette expérience, un bruit gaussien centré d'écart-type d'un demi pixel a été ajouté aux coordonnées des points dans l'image.

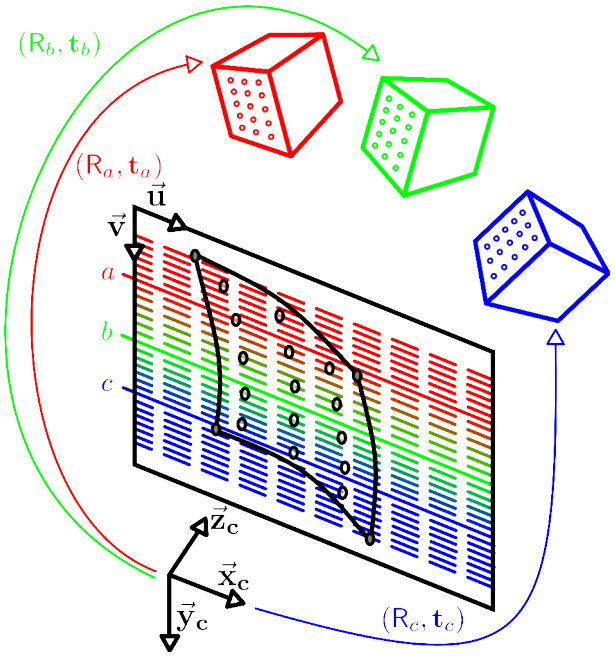


FIGURE IV.10 – Entre les lignes a et b, nous procédons à une interpolation des poses $(\mathsf{R}_a, \mathbf{t}_a)$ à $(\mathsf{R}_b, \mathbf{t}_b)$ grâce à une interpolation linéaire des paramètres de translation. Nous réalisons la même opération entre les lignes b et c.

---

1 Séparer les n points en correspondance en $\lfloor n/s \rfloor$ ensembles de s points consécutifs non-recouvrants ;
2 $\mathcal{M} \leftarrow \varnothing$; `// Liste des lignes moyennes de chaque ensemble`
3 **pour chaque** *Ensemble de points* **faire**
4 Calculer j, la ligne moyenne de l'ensemble ;
5 $\mathcal{M} \leftarrow \mathcal{M} \cup \{j\}$;
6 Estimer la pose moyenne $(\mathsf{R}_j, \mathbf{t}_j)$ en utilisant EPnP (Lepetit et al., 2009) avec les s points de l'ensemble ;
7 Interpoler linéairement $\{\mathbf{t}_j\}_{j=j_1}^{j_{\lfloor n/s \rfloor}}$ à partir de $\{\mathbf{t}_j\}_{j \in \mathcal{M}}$;
8 Interpoler $\{\mathsf{R}_j\}_{j=j_1}^{j_{\lfloor n/s \rfloor}}$ à partir de $\{\mathsf{R}_j\}_{j \in \mathcal{M}}$ en utilisant SLERP ;

FIGURE IV.11 – Algorithme d'estimation de la pose dynamique par la méthode d'uniformité interpolée par morceaux.

méthode «Global Shutter» par morceaux vient s'ajouter la complexité de cette méthode d'interpolation. Pour cette dernière, nous réalisons j_1 à $j_{\lfloor n/s \rfloor}$ interpolations qui sont des opérations de base. La complexité globale obtenue est donc de

$$\mathcal{O}(j_{\lfloor n/s \rfloor} - j_1) + \mathcal{O}(\lfloor n/s \rfloor s), \tag{IV.23}$$

or nous pouvons remarquer que en tout temps $j_{\lfloor n/s \rfloor} - j_1 < l$, où l est le nombre de lignes de l'image. Nous obtenons alors la complexité globale finale suivante

$$\mathcal{O}(l + n). \tag{IV.24}$$

Nous pouvons remarquer que généralement $n \ll l$, et que par conséquent la méthode d'initialisation est globalement quasi-linéaire par rapport au nombre de lignes de l'image.

CHAPITRE V

MODÉLISATION POLYNOMIALE DU «ROLLING SHUTTER»

Résumé

Comme il a été mentionné au chapitre précédent, l'une des difficultés des méthodes qui réalisent une optimisation locale de l'erreur de reprojection est de trouver une initialisation proche du minimum recherché. De la qualité de cette initialisation dépend grandement le temps nécessaire pour atteindre un minimum, mais aussi la précision du résultat final. De plus, nous n'avons aucune assurance qu'il s'agisse d'un résultat cohérent. En effet lorsqu'il n'est pas certain que les données correspondent au modèle, par exemple en présence de correspondances erronées, la convergence vers un minimum ne donne pas d'information sur l'éventuelle présence d'un autre minimum qui refléterait l'incohérence des données. L'échec de la convergence vers un minimum n'a pas d'interprétation évidente, en particulier cela n'assure pas qu'il n'existe pas un minimum malgré tout.

Dans le cadre de la robotique, ces différentes incertitudes sont problématiques et limitent l'utilisation des travaux présentés précédemment. En effet, pour contourner ces incertitudes, la solution retenue par (Dahmouche et al., 2008, 2009) a été d'imposer de commencer le suivi à un instant où l'objet n'est pas en mouvement. Une estimation de pose statique à cet instant permet alors de fournir une bonne initialisation pour l'instant suivant. Ensuite, plus loin dans la séquence, il est réalisé une prédiction à partir des images précédentes pour fournir l'initialisation pour l'image courante. En procédant ainsi de proche en proche, l'estimation peut être amenée à dévier légèrement petit à petit de la réalité, en raison de la présence de bruit notamment, mais aussi d'un changement brutal de mouvement. Il n'est donc pas garanti que sur le long terme l'optimisation ne finisse pas par échouer à nouveau lorsque l'erreur cumulée sur les images précédentes fausse la prédiction et fournit une initialisation de mauvaise

qualité.

Il serait donc intéressant de pouvoir disposer d'une méthode permettant de traiter le cas minimal tout en assurant de trouver le minimum global lorsqu'il existe, sans avoir recourt à une initialisation, et qui n'échoue que lorsque le modèle ne correspond pas aux données. Une telle méthode, en plus de résoudre les problèmes précédemment évoqués, ouvre aussi la voie à la mise en correspondance automatique. C'est ce que nous nous proposons d'étudier dans ce chapitre au travers des travaux publiés dans (Magerand et al., 2012).

V.1 Présentation du modèle

V.1.1 Origine et hypothèse du modèle

Bien que le modèle de projection de (Ait-Aider et al., 2006, 2007; Ait-Aider and Berry, 2009) soit non-linéaire, quelques remarques permettant de proposer une hypothèse simplificatrice méritent d'être faites. Tout d'abord nous pouvons remarquer que bien que l'erreur de reprojection du modèle perspectif soit par définition non-linéaire en raison du rapport à la profondeur, la non-linéarité du modèle «Rolling Shutter» uniforme est encore accentuée par la présence de fonctions trigonométrique dans le modèle du mouvement. Il est néanmoins possible de noter que ces fonctions trigonométriques ont toutes pour argument l'angle de la rotation par rapport à la rotation initiale à l'instant de l'acquisition d'une ligne de l'image. Il est donné par

$$\theta = t\omega, \tag{V.1}$$

où $t = \tau l$ est l'instant de l'acquisition de la ligne l avec τ le temps d'exposition d'une ligne et ω est la vitesse de rotation donnée en radian par secondes.

À condition que la vitesse d'acquisition des lignes de l'image soit suffisamment élevée, la variation de cet angle reste relativement faible, quand bien même la rotation se ferait à vitesse élevée. En effet si le temps séparant l'exposition des lignes est de l'ordre de la micro seconde (en pratique il peut même être inférieur), l'image fait tout au plus 3000 lignes sur une rétine de très haute résolution, ainsi l'instant de l'acquisition de la dernière ligne est au plus de l'ordre de quelques milli-secondes après celui de l'acquisition de la première ligne. Il faudrait alors une vitesse de rotation de plusieurs dizaines de milliers de tour par minute pour que l'angle de la rotation entre la première ligne et la dernière dépasse le dixième de radian.

À la vue de ces valeurs, il est raisonnable de supposer que la variation de cet angle est généralement suffisamment faible pour pouvoir utiliser les développements limités des fonctions trigonométriques.

V.1.2 Modèle de projection polynomial

En conservant les notations de la section III.3.2.1 et en écrivant le développement limité des fonctions trigonométriques à l'ordre premier avec l'hypothèse de la section précédente, nous obtenons que lorsque $t\omega \to 0$ alors $\cos(t\omega) \to 1$ et $\sin(t\omega) \to t\omega$. Ainsi la formule de Rodrigue donnée équation (III.2) à l'instant t devient

$$\delta\mathsf{R}(t, \mathbf{a}, \omega) \to \mathtt{I} + [\mathbf{a}]_\wedge \, t\omega. \tag{V.2}$$

Il est alors possible de poser $\mathbf{w} = \mathbf{a}\omega$ et de considérer qu'à l'instant t

$$\delta\mathsf{R}(t, \mathbf{w}) = \mathtt{I} + [\mathbf{w}]_\wedge \, t. \tag{V.3}$$

En notant que $\mathbf{a}$ est unitaire, nous pouvons retrouver

$$\omega = \|\mathbf{w}\| \text{ et } \mathbf{a} = \frac{\mathbf{w}}{\omega}. \tag{V.4}$$

Le modèle de projection d'un point $\mathbf{p}_i$ sur son image $\mathbf{m}_i = \begin{bmatrix} u_i & v_i \end{bmatrix}^\top$ à l'instant τ_i devient ainsi

$$\tilde{\mathbf{m}}_i \sim \mathsf{K}[\underbrace{\mathsf{R}_0 + \mathsf{R}_0 \tau_i \, [\mathbf{w}]_\wedge}_{=\mathsf{R}_i} \mid \underbrace{\mathbf{t}_0 + \tau_i \mathbf{v}}_{=\mathbf{t}_i}]\tilde{\mathbf{p}}_i. \tag{V.5}$$

Lorsque la rotation R_0 est paramétrée par un quaternion unitaire ou par la paramétrisation directe, cette matrice est constituée respectivement de

- neuf polynômes de degré deux, fonctions du quaternion ;
- ou de neuf monômes de degré un, fonctions des éléments de la matrice de rotation.

Le membre droit de l'équation (V.5) est donc un vecteur de trois polynômes fonctions des paramètres de la rotation, de $\mathbf{w}$, de $\mathbf{t}_0$ et de $\mathbf{v}$. Ce modèle de projection est donc polynomial.

Si nous supposons la caméra étalonnée, ce qui peut être réalisé avec des images statiques comme montré section II.3, alors la matrice K contenant les paramètres internes de la caméra est connue et elle est inversible. Il est alors possible de changer de repère et d'exprimer l'équation (V.5) dans le repère caméra et non plus dans le repère image selon

$$\mathsf{K}^{-1}\tilde{\mathbf{m}}_i \sim [\mathsf{R}_i|\mathbf{t}_i] \, \tilde{\mathbf{p}}_i. \tag{V.6}$$

Pour des raisons de stabilité numérique abordées dans Hartley and Zisserman (2003) et qui seront précisées section V.2.4, nous multiplions chaque membre du modèle de projection (V.6) par une matrice de normalisation des données

$$\mathsf{N} = \begin{bmatrix} \gamma & 0 & -\gamma\mu_u \\ 0 & \gamma & -\gamma\mu_v \\ 0 & 0 & 1 \end{bmatrix}. \tag{V.7}$$

Elle est composée d'une translation de l'origine vers le barycentre $\begin{bmatrix} \mu_u & \mu_v \end{bmatrix}^\top$ des projections et d'une mise à l'échelle isotropique de facteur γ tel que la distance moyenne des projections à cette origine soit de $\sqrt{2}$. Cela revient à faire un changement de repère vers un repère que nous désignerons comme étant le repère normalisé. En notant $\tilde{\mathbf{d}}_i = \mathsf{N}\mathsf{K}^{-1}\tilde{\mathbf{m}}_i = [f_i, g_i, 1]^\top$ les coordonnées de $\tilde{\mathbf{m}}_i$ exprimées dans le

repère normalisé, nous arrivons alors à

$$\tilde{\mathbf{d}}_i \sim \mathsf{N}\,[\mathsf{R}_i|\mathbf{t}_i]\,\tilde{\mathbf{p}}_i. \tag{V.8}$$

V.1.3 Erreur algébrique

L'erreur de reprojection géométrique qui est définie section II.3.2 est aisément interprétable géométriquement : c'est la distance euclidienne entre la projection mesurée et la projection estimée. Néanmoins avec le modèle de projection «Rolling Shutter» polynomial de l'équation (V.5), le calcul de l'erreur de reprojection introduit un ratio de polynômes. Afin d'éviter cela et pour aboutir à un système d'équation polynomial, une autre mesure de l'erreur doit être utilisée. L'équation (V.8) peut s'interpréter comme le fait que les vecteurs correspondants à chacun des deux membres soient colinéaires. Deux vecteurs colinéaires ont pour propriété d'avoir un produit vectoriel nul puisque la norme du produit vectoriel de deux vecteurs $\mathbf{a}$ et $\mathbf{b}$ dépend directement du sinus de l'angle θ entre eux selon

$$\|[\mathbf{a}]_\wedge\,\mathbf{b}\| = \|\mathbf{a}\|\|\mathbf{b}\|\sin(\theta). \tag{V.9}$$

Lorsque les vecteurs sont colinéaires, ils forment un angle θ qui est nul modulo π, nous avons alors $\sin(\theta) = 0$. Par conséquent la norme du produit vectoriel est nulle et donc le produit vectoriel est le vecteur nul. Dans notre cas, les deux vecteurs correspondant aux deux membres de l'équation (V.8) sont différents du vecteur nul et leurs normes sont donc non-nulles. Minimiser le produit vectoriel entre eux implique de minimiser sa norme et revient donc à minimiser l'angle θ qu'ils forment, tendant ainsi à les rendre colinéaires comme illustré dans la figure V.1. Il est ainsi possible d'introduire le vecteur d'erreur pour un point selon

$$\mathbf{h}_i = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}}_{=\mathsf{S}} \left[\tilde{\mathbf{d}}_i\right]_\wedge \mathsf{N}[\mathsf{R}_i|\mathbf{t}_i]\tilde{\mathbf{p}}_i = \mathbf{0}. \tag{V.10}$$

La matrice S sert à éliminer la troisième ligne du produit vectoriel. Celle-ci est une combinaison linéaire des deux premières et n'apporte donc aucune information utile. L'erreur algébrique pour un point est alors donnée par

$$\|\mathbf{h}_i\|. \tag{V.11}$$

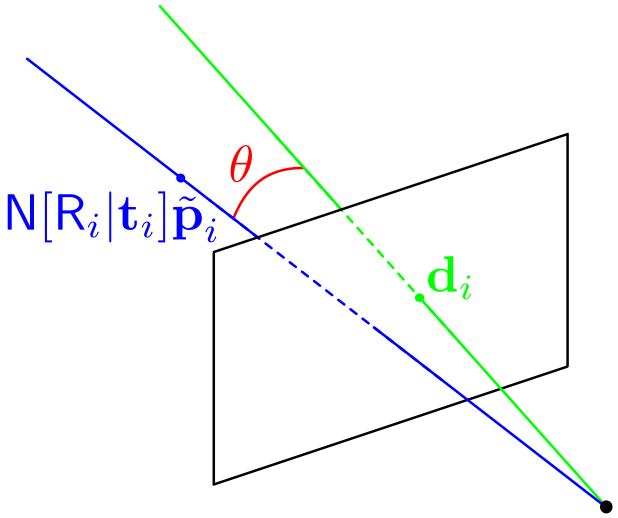


FIGURE V.1 – Minimiser l'erreur algébrique revient à minimiser le produit vectoriel $\left[\tilde{\mathbf{d}}_i\right]_\wedge \mathsf{N}[\mathsf{R}_i|\mathbf{t}_i]\tilde{\mathbf{p}}_i$. Sa norme est ainsi minimisée et comme $\mathbf{d}_i$ et $\mathsf{N}[\mathsf{R}_i|\mathbf{t}_i]\tilde{\mathbf{p}}_i$ sont des vecteurs non-nuls, nous minimisons donc l'angle θ entre eux. Ces deux vecteurs sont donc rapprochés l'un de l'autre jusqu'à être éventuellement colinéaires.

V.2 Estimation des paramètres

V.2.1 Fonction de coût

Une fois l'erreur de reprojection polynomiale définie pour un point, il reste à trouver le minimum de celle-ci pour l'ensemble des n points, ce qui forme le système polynomial

$$\begin{cases} \mathbf{h}_1 = \mathbf{0} \\ \vdots \\ \mathbf{h}_n = \mathbf{0}, \end{cases} \tag{V.12}$$

sous les contraintes suivantes nécessaires pour que la représentation choisie de la rotation initiale soit consistante :

$$\begin{cases} {\mathsf{R}_0}^\top \mathsf{R}_0 = \mathtt{I}_3 \\ |\mathsf{R}_0| = 1 \end{cases} \qquad \text{ou} \qquad \|\mathbf{q}_0\| = 1. \tag{V.13}$$

Résoudre directement ce système de manière exacte ne donnerait pas la solution recherchée puisqu'en raison du bruit, les différentes erreurs de reprojection ne seront jamais exactement nulles. De plus ce système a un degré total qui est de deux ou trois selon la paramétrisation de la rotation initiale utilisée, pour respectivement 18 ou 13 variables. Un tel système possède un espace de solutions de très grande taille (2^{18} dans le premier cas, 3^{13} dans le deuxième), ce qui le place hors de portée des méthodes de résolution polynomiales actuelles qui rencontrent des difficultés d'ordre technique (accumulation d'erreurs d'arrondis principalement) lorsqu'il existe plus de quelques centaines de solutions.

Pour ces raisons, nous chercherons plutôt à optimiser la fonction de coût définie par

$$h = \sum_{i=1}^{n} \|\mathbf{h}_i\|^2. \tag{V.14}$$

Le degré de cette fonction de coût est de quatre ou six selon la paramétrisation choisie pour la rotation initiale. Les contraintes à imposer sont respectivement de degré trois ou deux. L'optimisation de cette fonction de coût peut être réalisée à partir des méthodes d'optimisation polynomiale décrites dans la section A.2. Ces méthodes présentent deux avantages : les contraintes sont exactement satisfaites et le ou les minima globaux sont trouvés et certifiés s'ils existent.

V.2.2 Élimination des paramètres de translation

Dans le système (V.12), les paramètres de translation (position initiale $\mathbf{t}_0$ et vitesse $\mathbf{v}$) ne sont présents dans l'erreur de reprojection que dans des monômes de degré un. Par conséquent, ils sont solutions d'un système linéaire par rapport aux paramètres de rotations, et peuvent donc être éliminés du système et donc de la fonction de coût optimisée.

Les neuf entrées de la matrice de rotation $\mathsf{R}_i = \begin{bmatrix} \mathbf{r}_i^1 & \mathbf{r}_i^2 & \mathbf{r}_i^3 \end{bmatrix}$ à l'instant de la projection du point $\mathbf{p}_i$ sont des polynômes des paramètres de rotations uniquement. Vectorisons cette matrice colonne par colonne et notons le résultat

$$\boldsymbol{\rho}_i = \begin{bmatrix} \mathbf{r}_i^1 \\ \mathbf{r}_i^2 \\ \mathbf{r}_i^3 \end{bmatrix}. \tag{V.15}$$

Il est possible de remarquer que $\mathbf{t}_i$, la translation à ce même instant peut être réécrit depuis sa définition selon

$$\mathbf{t}_i = \mathsf{C}_i \begin{bmatrix} \mathbf{t}_0 \\ \mathbf{v} \end{bmatrix}, \tag{V.16}$$

avec

$$\mathsf{C}_i = [\mathtt{I}_3 | \tau_i \mathtt{I}_3]\,. \tag{V.17}$$

L'erreur de reprojection algébrique définie équation (V.10) peut alors être écrite sous forme d'une matrice de coefficients multipliée par un vecteur de monômes sous la forme

$$\mathbf{h}_i = \mathsf{A}_i \boldsymbol{\rho}_i + \mathsf{B}_i \mathsf{N} \mathbf{t}_i = \mathbf{0}, \tag{V.18}$$

en définissant les matrices suivantes :

$$\mathsf{B}_i = \begin{bmatrix} 0 & -1 & g_i \\ 1 & 0 & -f_i \end{bmatrix}, \tag{V.19}$$

de taille 2×3 ;

$$\mathsf{A}_i = [x_i \mathsf{B}_i \mathsf{N} \mid y_i \mathsf{B}_i \mathsf{N} \mid z_i \mathsf{B}_i \mathsf{N}]\,, \tag{V.20}$$

de taille 2×9 .

Il est possible de réécrire l'équation (V.18) comme

$$\mathsf{B}_i \mathsf{N} \mathsf{C}_i \begin{bmatrix} \mathbf{t}_0 \\ \mathbf{v} \end{bmatrix} = -\mathsf{A}_i \boldsymbol{\rho}_i. \tag{V.21}$$

En considérant $n \geq 3$ points en correspondance, les paramètres de translation $\mathbf{t}_0$ et $\mathbf{v}$ sont alors solu-

tions du système linéaire donné par

$$\mathsf{D}\begin{bmatrix}\mathbf{t}_0\\ \mathbf{v}\end{bmatrix} = -\mathsf{A}\boldsymbol{\rho}, \tag{V.22}$$

où

- D est la matrice de taille $2n \times 6$ formée de la concaténation verticale des $\{\mathsf{D}_i = \mathsf{B}_i\mathsf{N}\mathsf{C}_i\}_{i\in[1;n]}$:

$$\mathsf{D} = \begin{bmatrix}\mathsf{B}_1\mathsf{N}\mathsf{C}_1\\ \vdots\\ \mathsf{B}_n\mathsf{N}\mathsf{C}_n\end{bmatrix}, \tag{V.23}$$

- A est la matrice $2n\times 9n$ bande-diagonale constituée de la concaténation diagonale des $\{\mathsf{A}_i\}_{i\in[1;n]}$:

$$\mathsf{A} = \begin{bmatrix}\mathsf{A}_1 & 0 & 0\\ 0 & \ddots & 0\\ 0 & 0 & \mathsf{A}_n\end{bmatrix}, \tag{V.24}$$

- $\boldsymbol{\rho}$ est le vecteur de taille $9n$ contenant la concaténation verticale des $\{\boldsymbol{\rho}_i\}_{i\in[1;n]}$:

$$\boldsymbol{\rho} = \begin{bmatrix}\boldsymbol{\rho}_1\\ \vdots\\ \boldsymbol{\rho}_n\end{bmatrix}. \tag{V.25}$$

Ce système est de rang plein lorsqu'au moins trois des points se projettent sur des lignes différentes en raison de la structure de D. En effet, de par sa définition, les lignes de la matrice B_i forment une famille libre [1] de $\mathbb{R}^3$. De par la forme de N, les lignes du produit $\mathsf{B}_i\mathsf{N}$ sont toujours une famille libre. La matrice C_i est constituée de deux blocs diagonaux, le premier étant la matrice identité $\mathtt{I}_3$ et le deuxième $\tau_i\mathtt{I}_3$. Les matrices D_i sont donc chacune constituées de deux lignes formant une famille libre de $\mathbb{R}^6$. Si au moins trois points se projettent sur des lignes différentes, il existera trois valeurs de τ_i différentes, et les six lignes des trois matrices D_i correspondantes formeront une famille libre de $\mathbb{R}^6$. Lorsque cette hypothèse est vérifiée, ce qui est généralement le cas, il est alors possible d'écrire

$$\begin{bmatrix}\mathbf{t}_0\\ \mathbf{v}\end{bmatrix} = -\mathsf{D}^{+}\mathsf{A}\boldsymbol{\rho}, \tag{V.26}$$

où D^{+} est la pseudo-inverse de Moore-Penrose calculée à partir de la décomposition en valeurs singulières de D comme décrit dans l'annexe B. L'erreur de reprojection pour le point $\mathbf{p}_i$ devient alors

1. Une famille libre est ensemble de vecteurs linéairement indépendants, c'est-à-dire qu'aucun n'est combinaison linéaire des autres.

$$\mathbf{h}_i = \mathsf{A}_i \boldsymbol{\rho}_i - \mathsf{B}_i \mathsf{N} \mathsf{C}_i \mathsf{D}^+ \mathsf{A} \boldsymbol{\rho}. \tag{V.27}$$

Les inconnues sont les paramètres de la rotation initiale et le vecteur vitesse de rotation qui forment les polynômes contenus dans $\boldsymbol{\rho}$ et qui sont détaillés dans la section suivante.

V.2.3 Monômes et matrices de coefficients

Nous présentons ici les monômes et les matrices de coefficients qui interviennent pour former la fonction de coût finale optimisée. Une fois les paramètres de translation éliminés, le système polynomial (V.12) écrit sous forme matricielle devient

$$(\mathsf{A} - \mathsf{D}\mathsf{D}^+\mathsf{A})\, \boldsymbol{\rho} = \mathbf{0}. \tag{V.28}$$

Le vecteur $\boldsymbol{\rho}$ contient des polynômes, et peut être réécrit comme étant le produit d'une matrice de coefficients et d'un vecteur de monômes. Le système (V.28) peut donc lui aussi être réécrit comme un produit d'une matrice de coefficient dépendant uniquement des données et d'un vecteur de monômes des paramètres. Il faut noter cependant que les polynômes contenus dans $\boldsymbol{\rho}$ dépendent de la paramétrisation choisie pour la rotation initiale, et que donc le vecteur de monômes dépend également de ce choix.

V.2.3.1 Paramétrisation directe

Le vecteur $\boldsymbol{\rho}$ est constitué de la concaténation verticale des n vecteurs $\boldsymbol{\rho}_i$ de taille neuf. Écrire les vecteurs $\boldsymbol{\rho}_i$ sous la forme d'un produit de matrice et d'un vecteur de monômes est aisé avec la paramétrisation directe de la rotation initiale.

En notant $\mathsf{R}_0 = \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 \end{bmatrix}$ et par définition de $\boldsymbol{\rho}_i$, nous obtenons facilement que

$$\boldsymbol{\rho}_i = \begin{bmatrix} \mathbf{r}_1 \\ \mathbf{r}_2 \\ \mathbf{r}_3 \end{bmatrix} + \tau_i \begin{bmatrix} w_3\mathbf{r}_2 - w_2\mathbf{r}_3 \\ w_1\mathbf{r}_3 - w_3\mathbf{r}_1 \\ w_2\mathbf{r}_1 - w_1\mathbf{r}_2 \end{bmatrix}, \tag{V.29}$$

ce qui peut aussi s'écrire

$$\boldsymbol{\rho}_i = \underbrace{\begin{bmatrix} \mathsf{I}_9 \mid \mathsf{G}_i \end{bmatrix}}_{=\mathsf{E}_i} \boldsymbol{\sigma}. \tag{V.30}$$

Le vecteur $\boldsymbol{\sigma}$ est le vecteur contenant les 27 monômes de degré deux intervenant dans l'ensemble des

$\boldsymbol{\rho}_i$, il s'écrit

$$\boldsymbol{\sigma} = \begin{bmatrix} \mathbf{r}_1 \\ \mathbf{r}_2 \\ \mathbf{r}_3 \\ w_1\mathbf{r}_2 \\ w_1\mathbf{r}_3 \\ w_2\mathbf{r}_1 \\ w_2\mathbf{r}_3 \\ w_3\mathbf{r}_1 \\ w_3\mathbf{r}_2 \end{bmatrix}. \tag{V.31}$$

La matrice G_i est définie par

$$\mathsf{G}_i = \tau_i \begin{bmatrix} 0_3 & 0_3 & 0_3 & -\mathrm{I}_3 & 0_3 & \mathrm{I}_3 \\ 0_3 & \mathrm{I}_3 & 0_3 & 0_3 & -\mathrm{I}_3 & 0_3 \\ -\mathrm{I}_3 & 0_3 & \mathrm{I}_3 & 0_3 & 0_3 & 0_3 \end{bmatrix}. \tag{V.32}$$

Notons E la concaténation verticale des n matrice E_i, le vecteur $\boldsymbol{\rho}$ s'écrit alors

$$\boldsymbol{\rho} = \mathsf{E}\boldsymbol{\sigma}. \tag{V.33}$$

Nous pouvons ainsi réécrire (V.28) selon

$$\underbrace{(\mathsf{A} - \mathsf{D}\mathsf{D}^{+}\mathsf{A})\mathsf{E}}_{=\mathsf{Q}}\boldsymbol{\sigma} = \mathbf{0}. \tag{V.34}$$

La fonction de coût finalement optimisée est donc

$$h = \boldsymbol{\sigma}^{\top}\mathsf{Q}^{\top}\mathsf{Q}\boldsymbol{\sigma}. \tag{V.35}$$

Cette écriture pourrait être modifiée pour obtenir le produit d'une matrice de coefficient et un nouveau vecteur de monômes de degré quatre, néanmoins cette forme ne présente aucun intérêt contrairement à celle de l'équation (V.35) dont certaines méthodes d'optimisation peuvent tirer parti puisque $\mathsf{Q}^{\top}\mathsf{Q}$ est une matrice symétrique.

V.2.3.2 Paramétrisation par quaternions

Le résultat avec la paramétrisation par quaternions peut être déduit du résultat obtenu avec la paramétrisation directe. Il suffit en effet de substituer dans $\boldsymbol{\sigma}$ les vecteurs $\mathbf{r}_1$, $\mathbf{r}_2$ et $\mathbf{r}_3$ par des vecteurs

contenant les polynômes contenus dans l'annexe (C.9). Il est ainsi possible d'obtenir

$$\mathbf{r}_1 = \begin{bmatrix} 1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & -2 \end{bmatrix} \begin{bmatrix} q_1^2 \\ q_2^2 \\ q_3^2 \\ q_4^2 \\ q_1 q_4 \\ q_2 q_3 \\ q_2 q_4 \\ q_1 q_3 \end{bmatrix}, \tag{V.36}$$

$$\mathbf{r}_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 2 & -2 & 0 & 0 \\ 1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 2 \end{bmatrix} \begin{bmatrix} q_1^2 \\ q_2^2 \\ q_3^2 \\ q_4^2 \\ q_2 q_3 \\ q_1 q_4 \\ q_1 q_2 \\ q_3 q_4 \end{bmatrix}, \tag{V.37}$$

$$\mathbf{r}_3 = \begin{bmatrix} 0 & 0 & 0 & 0 & 2 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & -2 \\ 1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} q_1^2 \\ q_2^2 \\ q_3^2 \\ q_4^2 \\ q_1 q_3 \\ q_2 q_4 \\ q_3 q_4 \\ q_1 q_2 \end{bmatrix}. \tag{V.38}$$

Le vecteur $\boldsymbol{\sigma}$ peut donc être écrit sous la forme d'une matrice, qui fait intervenir les trois matrices précédentes, et d'un vecteur de monômes, qui contient les dix monômes précédents ainsi que leurs produits par w_1, w_2 et w_3.

V.2.4 Conditionnement du problème

Dans la section V.1.2, nous avons introduits une matrice N de normalisation des données sur les coordonnées des projections dans le repère caméra. Lors du calcul des matrices de coefficients, ces co-

ordonnées sont utilisées comme entrée directe (dans les matrices B_i par exemple) ou sont multipliées par les coordonnées des points de la scène ou de l'objet (comme dans les matrices A_i). Or ces coordonnées ont un ordre de grandeur faible puisqu'elles sont de l'ordre du millimètre, elles ont donc tendance à dégrader le conditionnement de la matrice des coefficients finale. C'est pour éviter cela que la matrice de normalisation N a été introduite. Nous pouvons noter qu'elle est insérée de sorte à ne pas impacter le modèle.

Idéalement il aurait fallu faire également une normalisation des coordonnées des points de l'objet ou de la scène. Néanmoins ces coordonnées ont un ordre de grandeur nettement supérieur et ont un impact réduit sur le conditionnement de la matrice de coefficients. De plus cette normalisation n'est pas aussi aisément réalisée que celle des coordonnées des projections, puisque pour l'introduire il faudrait modifier le modèle. C'est pour ces raisons qu'elle n'est pas effectuée.

En revanche il existe une dernière source de détérioration significative du conditionnement de la matrice des coefficients, c'est l'ordre de grandeur des τ_i. Celui-ci est en effet extrêmement faible, de l'ordre de quelques centaines de nano-secondes. Il est donc nécessaire de normaliser également l'échelle de temps. Nous réalisons cette opération en décalant l'origine du temps de Δ pour l'amener à l'instant de la première projection puis en multipliant $\{\tau_i\}_{i\in[1;n]}$ par l'inverse de leur moyenne λ. Cela n'a pas d'impact sur le modèle, mais décale la pose initiale pour être celle à l'instant de la première projection et modifie l'unité dans laquelle sont exprimés les paramètres de vitesse (ω et $\mathbf{v}$), ce qui est facilement réversible : il suffit de multiplier les paramètres de vitesse par λ et de ramener la pose initiale à l'instant $-\Delta$ en utilisant (III.2).

V.2.5 Algorithme de l'estimation

L'objectif de l'algorithme présenté ici est de réaliser l'estimation de la pose initiale $(\mathsf{R}_0, \mathbf{t}_0)$ d'un objet dans la scène filmée par une caméra «Rolling Shutter» ainsi que son mouvement uniforme paramétré par $\mathbf{a}$, représentant l'axe de rotation, la vitesse de rotation ω et la vitesse de translation $\mathbf{v}$. Les données nécessaires pour cela sont :

- la matrice d'étalonnage K des paramètres internes de la caméra ;
- le temps τ séparant l'acquisition de deux lignes consécutives de l'image ;
- un modèle tridimensionnel de l'objet filmé sous la forme de points $\{\mathbf{p}_i\}_{i\in[1;n]}$;
- les coordonnées des projections de ces points dans l'image $\{\mathbf{m}_i = \begin{bmatrix} u_i & v_i \end{bmatrix}^\top\}_{i\in[1;n]}$.

Dans le cas de la paramétrisation directe de la matrice R_0, l'algorithme complet est présenté dans la figure V.2.

1 Calculer les instants de projection de chacun des points du modèle $\{\tau_i = v_i\tau\}_{i\in[1;n]}$;
2 Décaler l'origine de l'échelle de temps des instants de projection pour l'amener à $\Delta = \tau_1$;
3 Normaliser les instants de projection en les multipliant par $\lambda = \frac{1}{\text{mean}(\{\tau_i - \Delta\}_{i\in[1;n]})}$;
4 Calculer les coordonnées des projections exprimées dans le repère caméra $\{\mathsf{K}^{-1}\tilde{\mathbf{m}}_i\}_{i\in[1;n]}$;
5 Calculer la matrice de normalisation N ;
6 Calculer les coordonnées des projections exprimées dans le repère caméra normalisé $\{\tilde{\mathbf{d}}_i = \mathsf{N}\mathsf{K}^{-1}\tilde{\mathbf{m}}_i\}_{i\in[1;n]}$;
7 Calculer les matrices D, A et E d'après les équations (V.23) et (V.24) ;
8 Calculer la matrice $\mathsf{Q} = (\mathsf{A} - \mathsf{D}\mathsf{D}^{+}\mathsf{A})\mathsf{E}$;
9 Former le vecteur des monômes $\boldsymbol{\sigma}$ à partir des paramètres de $\mathbf{w} = \omega\mathbf{a}$ et R_0 ;
10 Former la fonction de coût $h = \boldsymbol{\sigma}^{\top}\mathsf{Q}^{\top}\mathsf{Q}\boldsymbol{\sigma}$;
11 Former les contraintes ${\mathsf{R}_0}^{\top}\mathsf{R}_0 = \mathtt{I}_3$ et $|\mathsf{R}_0| = 1$;
12 Réaliser l'optimisation polynomiale $\min\limits_{(\mathsf{R}_0,\mathbf{w})} h \quad s.c. \quad \begin{cases} {\mathsf{R}_0}^{\top}\mathsf{R}_0 = \mathtt{I}_3 \\ |\mathsf{R}_0| = 1 \end{cases}$;
13 Calculer les paramètres de translation $\begin{bmatrix} \mathbf{t}_0 \\ \mathbf{v} \end{bmatrix} = -\mathsf{D}^{+}\mathsf{A}\mathsf{E}\boldsymbol{\sigma}$;
14 Annuler l'effet de la normalisation du temps $\begin{bmatrix} \mathbf{w} \\ \mathbf{v} \end{bmatrix} = \lambda \begin{bmatrix} \mathbf{w} \\ \mathbf{v} \end{bmatrix}$;
15 Calculer la pose initiale $(\mathsf{R}_0, \mathbf{t}_0)$ à l'instant $-\Delta$ en utilisant (III.2) ;
16 Calculer $\omega = \|\mathbf{w}\|$ et $\mathbf{a} = \frac{\mathbf{w}}{\omega}$;

FIGURE V.2 – Algorithme de l'estimation de la pose initiale $(\mathsf{R}_0, \mathbf{t}_0)$ d'un objet dans la scène filmée par une caméra «Rolling Shutter» ainsi que son mouvement uniforme paramétré par $\mathbf{a}$, représentant l'axe de rotation, la vitesse de rotation ω et la vitesse de translation $\mathbf{v}$. Les données nécessaires pour cela sont : la matrice d'étalonnage K des paramètres internes de la caméra, le temps τ séparant l'acquisition de deux lignes consécutives de l'image, un modèle tridimensionnel de l'objet filmé sous la forme de points $\{\mathbf{p}_i\}_{i\in[1;n]}$, les coordonnées des projections de ces points dans l'image $\{\mathbf{m}_i = \begin{bmatrix} u_i & v_i \end{bmatrix}^{\top}\}_{i\in[1;n]}$.

V.3 Détails d'implémentation

V.3.1 Méthode d'optimisation polynomiale

Dans l'annexe A.2 deux méthodes globales d'optimisation polynomiale sont présentées : la méthode «Sums of Squares» et la méthode des moments. Le modèle présenté précédemment a été implémenté avec succès à l'aide de ces deux méthodes. Nous verrons ici les détails spécifiques à l'optimisation de chacune de ces implémentations ainsi que leurs avantages et inconvénients.

V.3.1.1 Méthode «Sums of Squares»

La première implémentation fonctionnelle a été réalisée en utilisant la méthode «Sums of Squares» et la boîte-à-outils SOSTOOLS (Prajna et al., 2004). C'est cette implémentation qui a permis la validation de l'hypothèse faite section V.1.1. Elle souffrait néanmoins de très nombreux défauts, le principal étant une lenteur de calcul déraisonnable : il fallait plusieurs minutes de calculs sur un ordinateur portable moderne [2] pour parvenir au résultat. Parmi les raisons qui ont conduit à cette lenteur, nous pouvons citer :

- Au sein de SOSTOOLS, tous les calculs sur les polynômes sont réalisés avec la boîte-à-outils de calcul formel de MATLAB 2007, qui bien que très complète n'est pas très performante.
- La paramétrisation de la rotation utilisée à l'époque était les quaternions, ce qui donne une fonction polynomiale de degré six à optimiser. La relaxation correspondante fait nécessairement appel à des monômes de plus haut degré. La complexité algorithmique étant exponentielle par rapport à ce degré, on comprend aisément le problème.
- La résolution du problème de programmation semi-définie sous-jacent était confiée à la boîte-à-outils SeDuMi (Sturm, 2001) qui est extrêmement stable numériquement mais avec un coût élevé au niveau de l'efficacité.

Afin d'améliorer la situation, nous avons suivi différentes approches, tant sur le modèle que sur son implémentation. Les plus significatives sont :

- Plusieurs révisions du modèle dans le but de diminuer légèrement le degré du polynôme de la fonction de coût optimisée.
- Analyse de l'impact du degré de relaxation et diminution en conséquence de celui-ci.
- Modification de SOSTOOLS pour utiliser une autre boîte-à-outils pour la résolution du problème de programmation semi-définie sous-jacent : SDPT3 (Toh et al., 1999). L'intérêt de ce point sera détaillé section V.3.2.

2. Par moderne, nous entendons une machine récente à l'époque de ces travaux, contenant donc un processeur quadri-coeur cadencé à 2, 6Ghz et une mémoire centrale de plusieurs giga-octets.

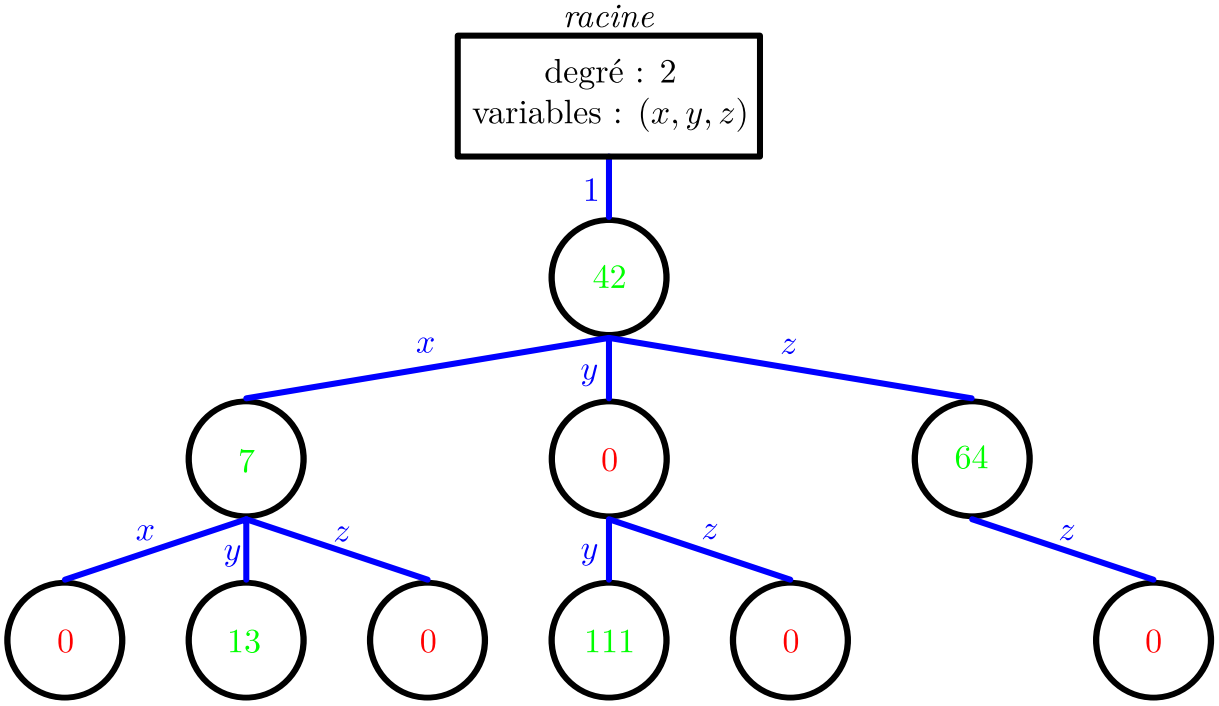


FIGURE V.3 – La structure d'arbre polynomial utilisée pour optimiser les calculs sur les polynômes. En noir se trouvent les noeuds qui contiennent chacun le coefficient, éventuellement nul, du monôme qu'il représente. En bleu se trouvent les arcs qui représentent les variables du problème, en suivant un arc, nous augmentons le degré de cette variable dans le monôme résultant. Ici le polynôme représenté est donc $42 + 7x + 13xy + 111y^2 + 64z$.

Une implémentation optimisée en langage C d'une structure pour la gestion des monômes et polynômes a également été réalisée, déchargeant ainsi une partie des calculs de la boîte-à-outils de calcul formel de MatLab. La structure utilisée est un arbre dont les noeuds correspondent aux monômes et contiennent le coefficient de ces derniers. Chaque noeud possède autant d'enfants qu'il y a de variables dans le problème, ils représentent les monômes de degré supérieur. Le noeud racine de l'arbre n'a qu'un seul enfant, le monôme 1 de degré 0. Il contient également des informations sur le polynôme : degré total et nombre de variables. La figure V.3 illustre cette structure de données. L'avantage majeur de cette structure est qu'il est très facile et très efficace de vérifier si un monôme existe déjà dans le polynôme, et de modifier son coefficient si besoin. Il suffit en effet de partir du noeud racine et de parcourir les variables du monôme par ordre lexicographique en descendant sur l'enfant correspondant. Par conséquent il devient trivial d'ajouter de nouveaux monômes, de réaliser les calculs de base sur les polynômes (addition et multiplication). Ces calculs sont très utilisés lors de l'étape qui forme la fonction de coût, les contraintes et la relaxation du problème.

Au final, lors des derniers tests avec cette méthode, le temps de calcul avait été divisé environ par deux, passant donc à un peu plus d'une minute sur une machine portable récente. Ce temps était difficilement plus compressible avec cette méthode, et toujours trop conséquent. De plus, début 2012, la version de MatLab nécessaire pour exécuter SOSTOOLS a cessé définitivement d'être supportée par MathWorks. Ce logiciel comportant un code qui empêche son utilisation cinq ans après sa parution, l'implémentation utilisant la méthode «Sums of Squares» a purement et simplement été abandonnée. De plus une alternative plus efficace avait de toute manière été trouvée, elle est détaillé dans la section suivante.

V.3.1.2 Méthode des moments

Bien que la première implémentation fonctionnelle ait été réalisée avec la boîte-à-outils SOSTOOLS, le peu d'efficacité dont elle faisait preuve a vite poussé à chercher une alternative plus viable, en plus d'une meilleure modélisation. La méthode des moments, qui est basée sur le problème dual de la méthode «Sums of Squares» (Lasserre, 2009; Laurent, 2009), semblait être un bon candidat.

Une première implémentation en reprenant la même modélisation, c'est-à-dire basée sur la paramétrisation par les quaternions de la rotation, divisait par deux le temps de calcul nécessaire pour réaliser une optimisation, le réduisant ainsi à quelques dizaines de secondes. Cette première implémentation était basée sur une version non modifiée de GloptiPoly (Henrion et al., 2009) utilisant SeDuMi pour la résolution du problème SDP sous-jacent. Une analyse de la répartition du temps de calcul comparée à celle de la méthode «Sums of Squares» montre que le gain se situe principalement dans la phase de préparation du problème de programmation semi-définie. Cela s'explique par le fait que la méthode des moments nécessite moins de calcul préparatoire, et de plus GloptiPoly réalise les calculs sur les polynômes avec une implémentation propre à cette boite à outils et indépendante du calcul formel de MatLab.

Un gain significatif a été obtenu en changeant la paramétrisation de la rotation pour utiliser la paramétrisation directe. Bien que cette paramétrisation demande d'imposer plus de contraintes, qui sont de degré deux ou trois, le degré de la fonction de coût est lui diminué pour passer à un degré quatre. Avec cette nouvelle paramétrisation, le temps nécessaire pour une optimisation est passé à une vingtaine de seconde en utilisant toujours SeDuMi pour la résolution du problème de programmation semi-définie. En utilisant toujours cette dernière paramétrisation et la boite à outils CSDP au lieu de SeDuMi, le temps final nécessaire pour une estimation est passé en dessous d'une dizaine de secondes.

V.3.2 Programmation semi-définie

Cette section présente les diverses boîtes à outils qui ont été utilisées lors des différentes implémentations du modèle polynomial. L'annexe A.2.1 présente le type de problème que peuvent résoudre ces boîtes à outils.

V.3.2.1 Boites à outils

Afin de résoudre le problème de programmation semi-définie résultant des deux méthodes utilisées pour optimiser la fonction de coût, de nombreuses boîtes à outils ont été testées. La plupart des librairies disponibles pour la résolution de ce type de problème sont d'installation souvent hasardeuse et pour un résultat parfois instable. Nous nous limiterons donc ici aux trois outils qui ont passé l'étape d'installation et satisfait quelques tests basiques pour vérifier leur stabilité.

En premier lieu, nous avons SeDuMi (Sturm, 2001) qui est la librairie utilisée par défaut tant par SOSTOOLS que par GloptiPoly. Elle est très simple d'installation, et est d'une robustesse à toute épreuve, ce qui en fait une des boîtes à outils les plus utilisées pour la résolution de ce type de problème. Sa stabilité a néanmoins un coût assez lourd en terme de performance puisqu'elle est implémentée principalement en MatLab avec seulement quelques mex-files pour optimiser quelques étapes critiques. Elle n'est de plus pas conçue pour utiliser le calcul parallèle alors même que la résolution de ce type de problème est massivement parallélisable. C'est néanmoins la première boîte à outils que nous avons utilisée pour valider le modèle «Rolling Shutter» polynomial.

Ayant constaté que la boîte-à-outils SeDuMi n'était pas optimisée autant qu'elle aurait pu l'être, il était nécessaire soit de l'optimiser, soit de la remplacer par une autre solution plus optimisée. Dans les librairies testées, la librairie SDPT3 (Toh et al., 1999) semblant satisfaire cette option puisqu'elle est intégralement implémentée sous forme de mex-file. SOSTOOLS a donc été modifiée pour pouvoir l'utiliser nativement en remplacement de SeDuMi. Elle présenta néanmoins un désavantage, celui d'être un peu moins stable numériquement que SeDuMi, ce qui se traduisit par des instabilités et des erreurs numériques lors de l'estimation des paramètres du modèle «Rolling Shutter» polynomial. De plus, le gain en performance n'était pas négligeable mais bien moindre que celui espéré. La raison est que, bien qu'étant optimisée par des mex-files, SDPT3 n'utilise pas non plus la parallélisation.

Ce sont ces diverses raisons qui ont poussé à chercher une meilleure alternative. Parmi les librairies testées, il en restait une prometteuse : CSDP (Borchers and Young, 2007). Cette dernière est intégralement implémentée en langage C et peut utiliser le calcul parallèle. Ce dernier aspect est détaillé dans la section suivante. Bien que CSDP soit fourni avec une interface MatLab, celle-ci passe par l'écriture sur le disque d'un fichier contenant les paramètres du problème de programmation semi-définie avant d'appeler l'exécutable qui va lire ce fichier et écrire dans un autre le résultat, lequel sera lu au retour

par l'interface MatLab pour extraire le résultat. Cette méthode n'est pas très efficace, et comme une réécriture de cette interface était rendue nécessaire pour utiliser la parallélisation, nous avons conçu une nouvelle interface sous forme de mex-file appelant directement la librairie.

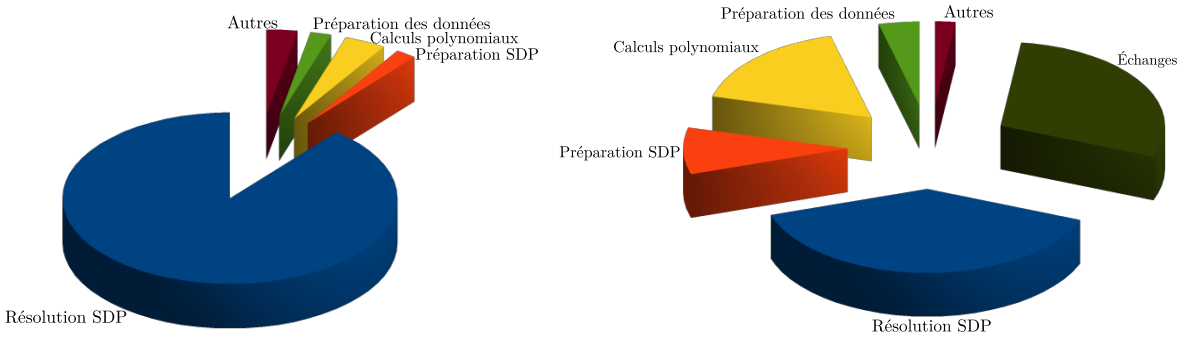
V.3.2.2 CSDP et parallélisation

Un travail non négligeable a dû être réalisé pour que la parallélisation soit pleinement opérationnelle, et de nombreux choix ont dû être faits. L'interface MatLab fournie d'origine avec CSDP ne permet pas de régler correctement les options nécessaires pour rendre le calcul parallèle le plus efficace possible, elle a donc été réécrite sous la forme d'un mex-file pour tirer parti au maximum de la parallélisation et rendre plus efficaces les opérations d'échange entre GloptiPoly et la librairie. La compilation même de CSDP a nécessité également de nombreux réglages pour utiliser correctement la parallélisation tout en restant stable numériquement.

Plusieurs éléments interviennent dans la parallélisation et nécessitent chacun d'être réglé, voire compilé, de sorte à rendre le tout le plus efficace possible :

- Une implémentation optimisée et parallélisée de LAPACK (Anderson et al., 1999) et BLAS (Dongarra et al., 1990), qui sont les librairies de référence pour tous les calculs algébriques. Ce sont les mêmes que celles utilisées au coeur de MatLab. L'implémentation ATLAS (Whaley et al., 2001) de ces deux outils a été utilisée, compilée en activant le support des threads.
- La boite à outils CSDP en elle même car compilée avec les options par défaut, elle n'utilise pas les capacités de calcul parallèle. Pour ce faire, il faut activer à la compilation le support de ATLAS et de l'extension OpenMP (OpenMP, 2000) du langage C. Cette dernière définit des structures de contrôle spécifique à la parallélisation.
- L'interface qui fait appel à CSDP doit régler des variables d'environnement spécifiques à OpenMP et ATLAS pour les activer et configurer certains aspects de la parallélisation comme le nombre de coeurs et de threads à utiliser.

Le choix des paramètres de parallélisation est primordial car cette dernière provoque certes un gain de temps au niveau des calculs mais nécessite également un travail supplémentaire de réunification d'autant plus important que nous séparons le problème en de multiples morceaux. Pour trouver la juste balance nécessaire pour obtenir l'efficacité maximum, une approche principalement expérimentale a été utilisée. Bien que quelques considérations théoriques puissent justifier le choix final, celui-ci reste de toute manière dépendant de la machine sur laquelle les calculs sont réalisés. Néanmoins il apparaît expérimentalement qu'utiliser plus de quatre unités de calcul n'est pas efficace au vu de la taille du problème de programmation semi-définie à résoudre, laquelle est constante quel que soit le nombre de points utilisés. De plus, le temps d'allocation des ressources nécessaires pour le calcul parallèle pénalise légèrement la première estimation lorsque nous en réalisons plusieurs. Les estimations suivantes



(a) Avec SeDuMi (total : 17.695 secondes).

(b) Avec CSDP (total : 9.385 secondes).

FIGURE V.4 – Comparaison de la répartition du temps de calcul avec l'implémentation utilisant SeDuMi et celle utilisant la version parallélisée de CSDP. La part du temps de calcul utilisée pour la résolution du problème de programmation semi-définie est nettement inférieure avec CSDP, et le temps total nécessaire est presque divisé par deux. Nous pouvons remarquer que sans interface optimisée pour les échanges entre MatLab et CSDP, la part du temps des entrées et sorties est considérable. Le temps total moyen indiqué est celui pour une estimation isolée.

n'ayant pas à allouer ces ressources, elles sont légèrement plus rapides. La figure V.4 montre le gain obtenu au niveau de la répartition du temps de calcul entre l'utilisation de SeDuMi et la version parallélisée de CSDP, sans l'interface d'échange optimisée.

V.3.3 Travaux futurs

Avec la méthode actuelle de résolution basée sur GloptiPoly et CSDP avec une interface optimisée, il semble impossible de pouvoir encore accélérer les calculs. C'est pour cette raison qu'il faudrait se tourner vers une nouvelle méthode de résolution tout en conservant le modèle utilisé. Une alternative à l'utilisation de GloptiPoly et CSDP serait de trouver les points stationnaires de la fonction de coût en résolvant le système obtenu en dérivant cette dernière par rapport aux différentes variables.

Ce système est très simple à obtenir, en effet le gradient de la fonction de coût est facilement calculé à partir de (V.35). La dérivé de h par rapport à chacune des variables est en effet donné par

$$h' = \boldsymbol{\sigma}^\top \mathrm{Q}^\top \mathrm{Q} \boldsymbol{\sigma}' + \boldsymbol{\sigma}'^\top \mathrm{Q}^\top \mathrm{Q} \boldsymbol{\sigma}, \tag{V.39}$$

où $\boldsymbol{\sigma}'$ est le vecteur des monômes dérivés par rapport à la variable considérée. La matrice $\mathrm{Q}^\top \mathrm{Q}$ étant symétrique, $\boldsymbol{\sigma}^\top \mathrm{Q}^\top \mathrm{Q} \boldsymbol{\sigma}' = \boldsymbol{\sigma}'^\top \mathrm{Q}^\top \mathrm{Q} \boldsymbol{\sigma}$ et la dérivé s'écrit donc plus simplement

$$h' = 2\boldsymbol{\sigma}^\top \mathrm{Q}^\top \mathrm{Q} \boldsymbol{\sigma}'. \tag{V.40}$$

En utilisant la paramétrisation directe de la rotation initiale, les douze équations ainsi obtenues et agrémentées des sept équations des contraintes de l'équation (V.13) forment un système de dix-neuf équations de degré au plus trois pour douze variables. L'espace de solution théorique est donc de 3^{12}, ce qui reste en dehors de la portée des méthodes de résolution actuelles de systèmes polynomiaux. Néanmoins, ce système d'équations est sur-contraint, et son nombre de degrés de liberté réel est de six. L'espace de solution correspondant est donc de 3^6, ce qui est à la portée de méthodes de résolution récentes telle que celle décrite dans (Byröd et al., 2009; Kukelova et al., 2008).

V.4 Mise en correspondance automatique

La garantie de convergence vers le minimum global de la méthode d'estimation présentée précédemment nous permet de réaliser la mise en correspondance automatique des points du modèle tridimensionnel et de leur projection lorsque les objets ou la scène considérés sont suffisamment texturés. Pour cela nous effectuons une estimation robuste des paramètres du modèle à partir de correspondances obtenues à l'aide de descripteurs de points tels que «SIFT» (Lowe, 2004) ou «SURF» (Bay et al., 2006) entre la nouvelle image et des images où les correspondances sont connues.

V.4.1 Obtention du modèle tridimensionnel

Si nous supposons les objets et la scène suffisamment texturés, alors il est possible d'obtenir un modèle tridimensionnel simplement en utilisant une paire stéréoscopique de caméras étalonnées. Une simple prise de vue permet d'obtenir un modèle d'une grande précision lorsque l'étalonnage a été effectué correctement. Les correspondances de points entre les deux vues peuvent être obtenues avec les descripteurs de points comme SIFT (Lowe, 2004) ou SURF (Bay et al., 2006). À partir des correspondances de points, le modèle tridimensionnel est estimé par triangulation. Les détails de la reconstruction de la structure à partir de la triangulation sur des images stéréo sont donnés dans l'annexe D.

V.4.2 Correspondances 2D-3D initiales

Pour réaliser l'estimation des paramètres du modèle, il est nécessaire de connaître des correspondances entre les points $\mathbf{p}_i$ et leurs projection $\mathbf{m}_i$. Lors d'expériences de validation du modèle, ces correspondances peuvent être obtenues de manière certaine en les réalisant à la main. Cependant pour être utilisable dans une application concrète, cette étape doit pouvoir être réalisée automatiquement.

Mettre en correspondance des points 3D avec leurs projections 2D n'est pas une tâche aisée. En effet, il est difficile de déduire à partir du modèle 3D d'un objet ou d'une scène des caractéristiques suffisantes pour différencier les projections 2D. Il est en revanche très facile de caractériser les projections 2D dans une image, et ensuite de comparer avec les projections dans une autre image. Il a été question de ce point dans la section II.4.

Une méthode simple pour utiliser des correspondances entre projections 2D dans différentes images pour retrouver les correspondances avec le modèle 3D consiste à utiliser une ou plusieurs images comme gabarit. Sur ces gabarits, les correspondances avec le modèle 3D auront été réalisées préalablement, que ce soit manuellement ou automatiquement. Lorsqu'une nouvelle image doit être mise en correspondance, nous comparons les descripteurs des points d'intérêt de celle-ci avec les descripteurs

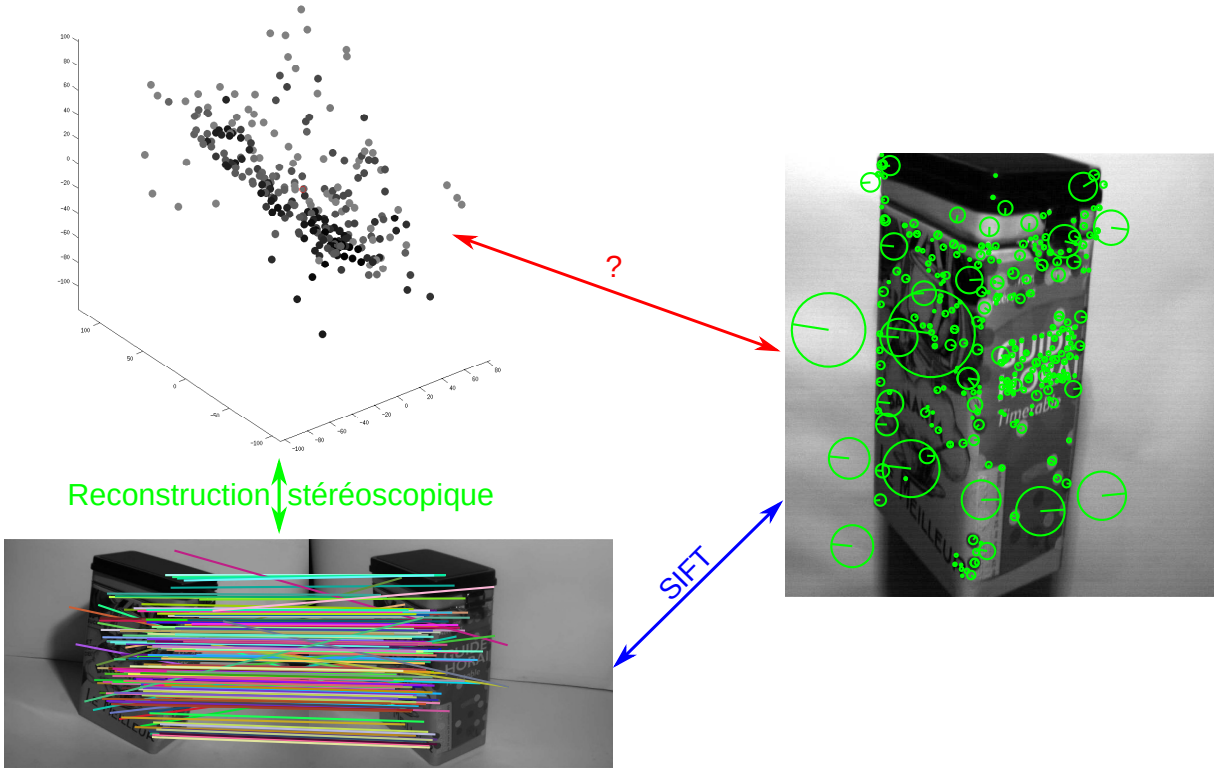


FIGURE V.5 – La mise en correspondances des points du modèle tridimensionnel avec les projections sur une image inconnue est réalisée via les images de la vue stéréoscopique utilisée pour reconstruire le modèle tridimensionnel. Les correspondances entre le modèle et les images de la vue stéréoscopique sont conservées lors de la reconstruction, permettant ainsi de les utiliser comme gabarit pour faire la mise en correspondance entre une image inconnue et le modèle tridimensionnel.

des gabarits, et nous conservons celui qui est le plus proche. Cela donne une correspondance candidate avec le modèle 3D pour la nouvelle image. Ce principe est illustré figure V.5.

Le choix des gabarits est particulièrement important, il faut bien évidemment que les points d'intérêt y soient correctement visibles et différentiables. Nous avons choisi de prendre comme gabarits les deux images utilisées lors de la reconstruction du modèle tridimensionnel. Les descripteurs des points dans ces deux images auront été conservés lors de celle-ci de sorte qu'il n'y ait plus qu'à calculer les descripteurs dans la nouvelle image et les comparer avec ceux sauvegardés.

V.4.3 Estimation robuste

Le détecteur et descripteur de points d'intérêt que nous avons utilisé pour mettre en correspondance les gabarits avec l'image courante est très robuste, mais il n'est cependant pas conçu pour être insensible aux déformations provoquées par le «Rolling Shutter». Par conséquent lors de la mise en correspondance, un certain taux des correspondances est erroné. Celui-ci n'est pas connu à l'avance, et dépend de la qualité des textures des objets et de l'importance des déformations induites par le «Rolling Shutter». De plus, le modèle tridimensionnel obtenu par la reconstruction stéréoscopique peut également contenir des points erronés, qui faussent eux aussi l'estimation. Il est alors préférable de réaliser une estimation robuste des paramètres en filtrant les correspondances erronées, par exemple en utilisant la méthode «RANSAC» qui est présentée section A.4. Bien que l'implémentation de «RANSAC» soit aisée, un certain nombre de points demandent ici une attention particulière.

En premier lieu, il faut noter que la construction de la matrice Q la rend totalement dépendante de l'ensemble des données qui ont été utilisées pour la calculer. En effet lors de l'élimination des paramètres de translation, nous réalisons une pseudo-inverse qui utilise l'ensemble des données. Il est donc impossible d'utiliser la matrice Q calculée pour l'ensemble des données pour estimer le modèle avec seulement un sous-ensemble des données. Il est nécessaire pour cela de recalculer Q pour chaque sous-ensemble de données.

Pour la même raison, il est impossible de retrouver l'erreur $\mathbf{h}_i$ définie équation (V.10) pour un point donné à partir de la matrice Q. Pour séparer les données erronées des données correctes, il est donc nécessaire de recalculer cette erreur à partir des paramètres estimés en utilisant directement (V.27). De plus, la norme de cette erreur est difficilement interprétable en raison du produit vectoriel. Il est donc compliqué de définir le seuil utilisé pour différencier les données correctes de celles erronées. Le recours direct à l'erreur de reprojection est une solution simple qui permet de résoudre ce problème.

Pour finir, le taux de données correctes dans les données d'entrée n'est pas connu et difficilement estimable par avance, il est donc impossible de fixer le nombre d'itérations nécessaires *a priori*. Pour contourner ce problème il est possible d'estimer ce taux au fur et à mesure des itérations en réalisant la moyenne du taux de données considérées comme correctes lors de la segmentation. Le nombre d'itérations est alors recalculé à chaque itération de manière dynamique en utilisant l'équation (A.25).

CHAPITRE VI

RÉSULTATS EXPÉRIMENTAUX

Résumé

Dans ce chapitre nous présenterons une comparaison des différentes méthodes proposées avec les travaux précédents (Ait-Aider et al., 2006). Pour cela nous nous intéresserons dans un premier temps à la réalisation d'un simulateur de données «Rolling Shutter». À l'aide de ce simulateur, nous avons créé deux jeux de données couvrant tous types de mouvements, notamment uniformes pour le premier et non-uniformes pour le second. Diverses expériences ont alors été réalisées pour étudier le comportement des différentes méthodes vis-à-vis du nombre de points utilisés pour réaliser l'estimation ou encore l'effet du bruit sur la précision de l'estimation. Quatre paramètres d'erreur ont été observés afin de mieux cerner les avantages et inconvénients de chacune des méthodes.

Il ressort de cette étude que pour toutes les méthodes la précision est d'autant meilleure que le nombre de points utilisés pour réaliser l'estimation est élevé. Nous constatons notamment une très forte progression de la précision de six à douze points pour les modèles uniformes. Pour les modèles non-uniformes, un très grand nombre de points est nécessaire pour obtenir un résultat correct. En ce qui concerne l'évolution de la précision vis-à-vis du bruit, nous constatons que bien que certaines y soient plus tolérantes que d'autres, toutes les méthodes sont particulièrement sensibles à un bruit trop élevé. Un bruit supérieur à quelques pixels donne rapidement des résultats aberrants. Ceci s'explique naturellement par le fait que les déformations dans l'image dues au mouvement sont généralement de l'ordre de quelques dizaines de pixels au plus, ajouter un bruit de quelques pixels représente donc une perturbation importante par rapport à ce qui est mesuré.

Nous présenterons enfin des applications sur deux séquences d'images réelles pour les méthodes

uniformes. Les objets utilisés dans ces deux séquences sont des mires sur lesquelles sont placés des marqueurs circulaires détectés et mis en correspondances avec le modèle tridimensionnel à la main. La méthode d'estimation robuste présentée section V.4 sera également testée sur deux séquences de mouvement réel avec des objets texturés et donc le modèle aura préalablement été reconstruit automatiquement par stéréoscopie.

VI.1 Simulateurs «Rolling Shutter»

Écrire un simulateur de données «Rolling Shutter» n'est pas aussi trivial que d'écrire un simulateur «Global Shutter». Pour projeter un point il est en effet nécessaire de connaître l'instant auquel il sera projeté afin de calculer la position et l'orientation du repère objet par rapport au repère caméra à cet instant. Ce paramètre n'étant pas connu à l'avance, il est nécessaire de l'estimer durant la simulation. Deux méthodes remplissant cet objectif sont présentées dans cette section. Il faut noter que ces deux méthodes ne gèrent pas les phénomènes d'occultation. Nous supposons que le modèle tridimensionnel de l'objet, constitué de n points $\mathbf{p}_i$, est connu. La matrice de calibrage K de la caméra virtuelle est également donnée.

VI.1.1 Basé sur l'erreur de reprojection

La méthode la plus intuitive pour réaliser cette simulation est sans doute d'inverser le problème de l'estimation de pose. En conservant les notations de la section IV.1.1, nous supposons connue la position de l'objet à tous les instants où une ligne j du capteur simulé est exposée. Nous pouvons la représenter par une pose dynamique :

$$\{(\mathsf{R}(j,\mathbf{x}),\mathbf{t}(j,\mathbf{x}))\}_{j\in[1;l]}\,, \tag{VI.1}$$

où $\mathbf{x}$ sont les paramètres définissant le mouvement et l est le nombre de lignes. Il s'agît alors d'estimer la position $\mathbf{q}_i = \begin{bmatrix} u_i & v_i \end{bmatrix}^\top$ de la projection des points $\mathbf{p}_i$ de l'objet, l'instant de cette projection étant défini par la ligne $j = v_i$. Nous réalisons donc pour chaque point $\mathbf{p}_i$ la minimisation suivante :

$$\min_{\mathbf{q}_i} \|\mathbf{q}_i - \boldsymbol{\varphi}\left(\mathsf{K}\begin{bmatrix} \mathsf{R}(v_i,\mathbf{x}) & \mathbf{t}(v_i,\mathbf{x}) \end{bmatrix}\tilde{\mathbf{p}}_i\right)\|^2, \tag{VI.2}$$

c'est une optimisation non-linéaire sans contrainte qui peut être réalisée avec la méthode de Levenberg-Marquardt décrite annexe A.3.1. L'initialisation de l'optimisation sera donnée par une projection «Global Shutter» avec la pose initiale ou la pose médiane. Bien que cette méthode de simulation soit très simple, elle présente néanmoins un désavantage en terme de réalisme de la projection «Rolling Shutter» : un point ne peut pas se projeter sur plusieurs lignes comme cela peut arriver en réalité. De plus il arrive que l'optimisation échoue, cet échec n'est pas interprétable et il est alors impossible de projeter le point correspondant.

Lire le modèle tridimensionnel de l'objet $\{\mathbf{p}_i\}_{i\in[1;n]}$;
Construire la pose dynamique $\{(\mathsf{R}(j,\mathbf{x}),\mathbf{t}(j,\mathbf{x}))\}_{j\in\mathcal{F}}$;
Initialiser un vecteur $\mathbf{q}$ de taille n rempli de listes vides ;
pour chaque *ligne* $j \in [1;l]$ **faire**
    pour chaque *point* $\mathbf{p}_i, i \in [1;n]$ **faire**
        Projeter le point $\mathbf{p}_i$ avec la pose $(\mathsf{R}(j,\mathbf{x}),\mathbf{t}(j,\mathbf{x}))$ sur $\mathbf{m}_i = \begin{bmatrix} u_i & v_i \end{bmatrix}^\top$;
        si $v_i = j$ **alors**
            $q_i = q_i \cup \mathbf{m}_i$;

FIGURE VI.1 – Algorithme du simulateur «Rolling Shutter» basé sur la projection «Global Shutter». Nous obtenons à la fin un vecteur de même taille que le nombre de points du modèle de l'objet, dont chaque entrée est une liste, éventuellement vide, des projections de ce point.

VI.1.2 Basé sur la projection «Global Shutter»

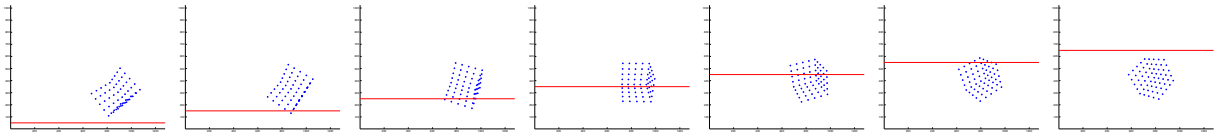
Une autre méthode de simulation qui est plus proche du processus réel de la projection «Rolling Shutter» est possible en s'inspirant de ce qui a été fait dans Forssén and Ringaby (2010). Pour cela nous considérons la projection «Global Shutter» à chaque instant où une ligne de la caméra simulée sera exposée, et nous ne conservons que les points qui se sont projetés sur la ligne correspondante. Il est alors possible qu'un point ne soit pas projetés, auquel cas nous savons qu'il aurait disparu sur une caméra réelle, mais aussi qu'il soit projeté sur plusieurs lignes. Pour gérer ces différentes situations, il est nécessaire pour chaque point de conserver une liste, éventuellement vide, de ses différentes projections. L'ensemble de ce mécanisme de projection est récapitulé dans l'algorithme VI.1 et est illustré figure VI.2.

VI.1.3 Jeux de données simulées

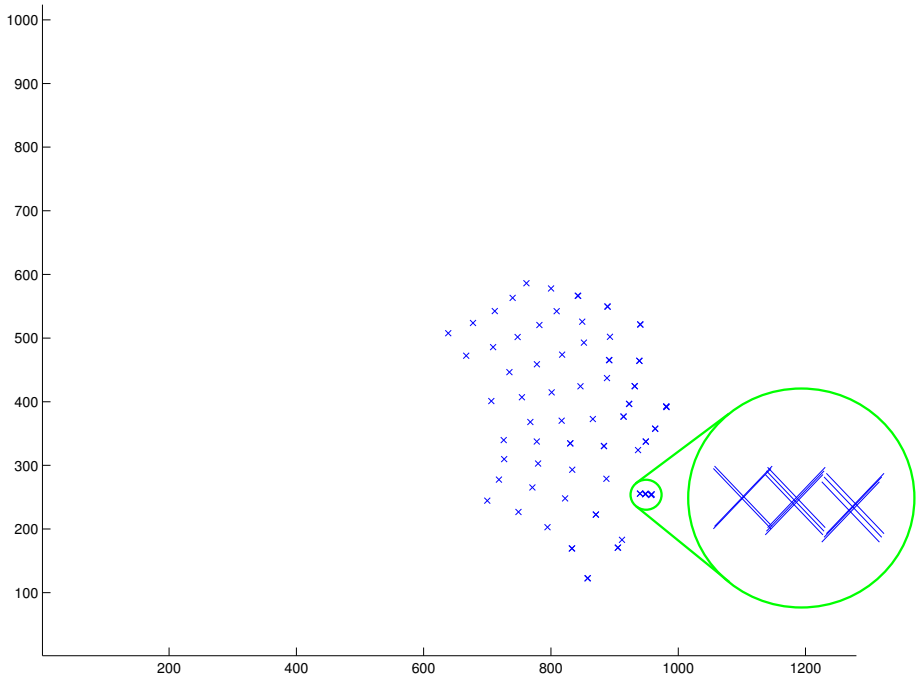
Deux jeux de données simulées ont été générés à partir des simulateurs présentés précédemment et seront utilisés pour comparer les résultats des deux méthodes étudiées, ils sont présentés ici.

VI.1.3.1 Avec un mouvement quelconque

Le premier jeu de données simulées a été généré à partir du simulateur basé sur l'erreur de reprojection. Il contient trente deux simulations de mouvements quelconques : translation pure, rotation pure, avec ou sans accélération. L'objet projeté dans ces simulations est constitué de 60 à 150 points répartis selon une grille sur trois faces d'un cube. Deux jeux de paramètres de caméras ont été utilisés, chacun pour la moitié des simulations. La figure VI.3 présente deux exemples d'images simulées obtenues. La répartition des projections le long des lignes de l'image est assez dense.

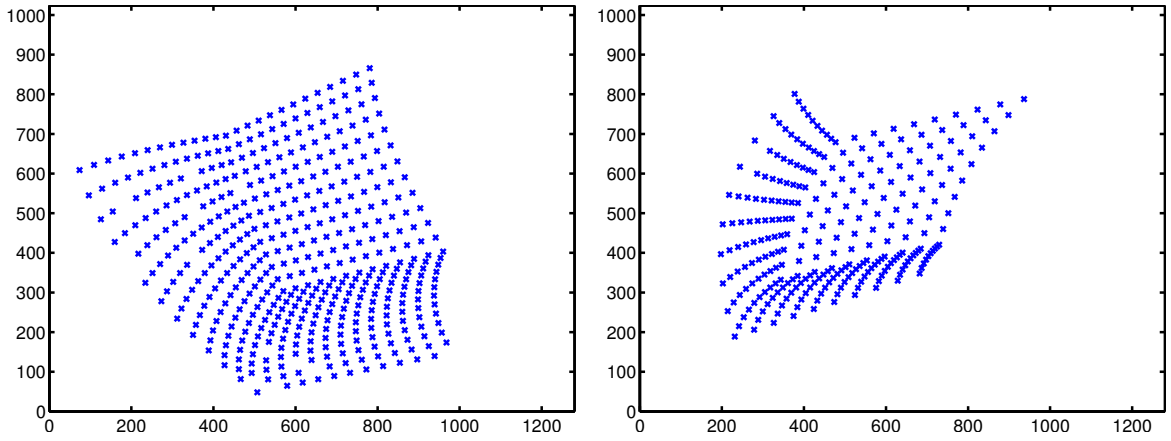


(a) Séquence «Global Shutter», la ligne rouge indique la ligne en cours d'exposition.



(b) Image finale, les points qui semblent apparaître en gras sont en fait des points qui se sont projetés sur plusieurs lignes.

FIGURE VI.2 – L'image finale de la simulation est obtenue en conservant dans chaque image de la séquence «Global Shutter» les points qui se projettent sur la ligne en cours d'exposition.



(a) Avec un mouvement de rotation pure non-uniforme.

(b) Avec un mouvement de rotation uniforme et de translation non-uniforme.

FIGURE VI.3 – Exemples d'images simulées obtenues avec un mouvement non-uniforme.

VI.1.3.2 Avec un mouvement uniforme

Le deuxième jeu de données simulées contient quarante images obtenues avec le simulateur basé sur la projection «Global Shutter». Pour la moitié de ces images l'objet projeté est constitué de 60 à 150 points répartis selon une grille sur trois faces d'un cube de taille variable. Pour l'autre moitié, l'objet projeté est constitué d'une centaine de points aléatoires dans un pavé de taille variable. À chacune des images les paramètres de la caméra sont changés et ajustés manuellement pour que les projections restent dans l'image. Les mouvements des objets projetés sont uniformes et représentent différents types de mouvements : rotation ou translation seule et mouvement mixte. En ce qui concerne la vitesse, nous avons séparé le jeu de données en deux catégories : mouvements relativement lents et mouvements très rapides.

VI.2 Résultats avec des données simulées

VI.2.1 Méthodologie

Les paramètres estimés par les différentes méthodes d'estimation de pose dynamique présentées ne sont pas directement comparables. Il est donc nécessaire de convertir tous ces résultats dans une paramétrisation commune pour pouvoir les comparer. Le plus aisé est de convertir les paramètres obtenus pour les méthodes utilisant un modèle de mouvement uniforme en une suite de poses définies à chaque instant d'exposition d'une ligne de l'image, qui est la paramétrisation résultante des méthodes avec modèle non-uniforme. Nous obtenons ainsi pour chaque modèle une pose dynamique telle que définie section IV.1.1 :

$$\{(\mathsf{R}_j, \mathbf{t}_j)\}_{j\in[1;l]}\,, \tag{VI.3}$$

où les rotations $\{\mathsf{R}_j\}_{j\in[1;l]}$ sont paramétrées par des quaternions unitaires $\{\mathbf{q}_j\}_{j\in[1;l]}$. Les données de la simulation sont également converties dans ce format.

Afin de comparer les données aux estimations obtenues par les différentes méthodes sur une image, nous nous intéressons pour chaque ligne $j \in [1; l]$ aux paramètres suivants :

- l'angle formé entre l'axe réel de la rotation et celui estimé ;
- la différence entre l'angle réel de la rotation et celui estimé ;
- l'angle formé entre le support de la translation réelle et celui de celle estimée ;
- la différence entre la norme de la translation réelle et celle estimée.

Pour une image donnée, l'erreur globale de chacun de ces paramètres est la racine de la moyenne quadratique le long des lignes. Dans le cas d'une séquence d'images, nous réalisons la moyenne des erreurs globales des différentes images.

Dans les légendes des figures de cette section, les acronymes suivants sont utilisés pour indiquer les différentes méthodes :

- «GS», le modèle «Global Shutter» en utilisant EPnP raffiné par une optimisation non-linéaire ;
- «URS», le modèle «Uniform Rolling Shutter» des travaux précédents tel que présentés section III.3.2.1, aussi appelé modèle uniforme ;
- «PWGS», le modèle «PieceWise Global Shutter» de l'initialisation présentée section IV.3, aussi appelé uniformité interpolée par morceaux ;
- «SLWRSn», le modèle «Scan-Line Wise Rolling Shutter» présenté section IV.1.4 initialisé grâce au modèle uniformité interpolée par morceaux, le chiffre n indiquant le degré auquel sont considérées les contraintes, ce modèle est aussi appelé modèle non-uniforme lisse de degré n ;
- «PURS», le modèle «Polynomial Uniform Rolling Shutter» présenté au chapitre V.

VI.2.2 Effet du nombre de points

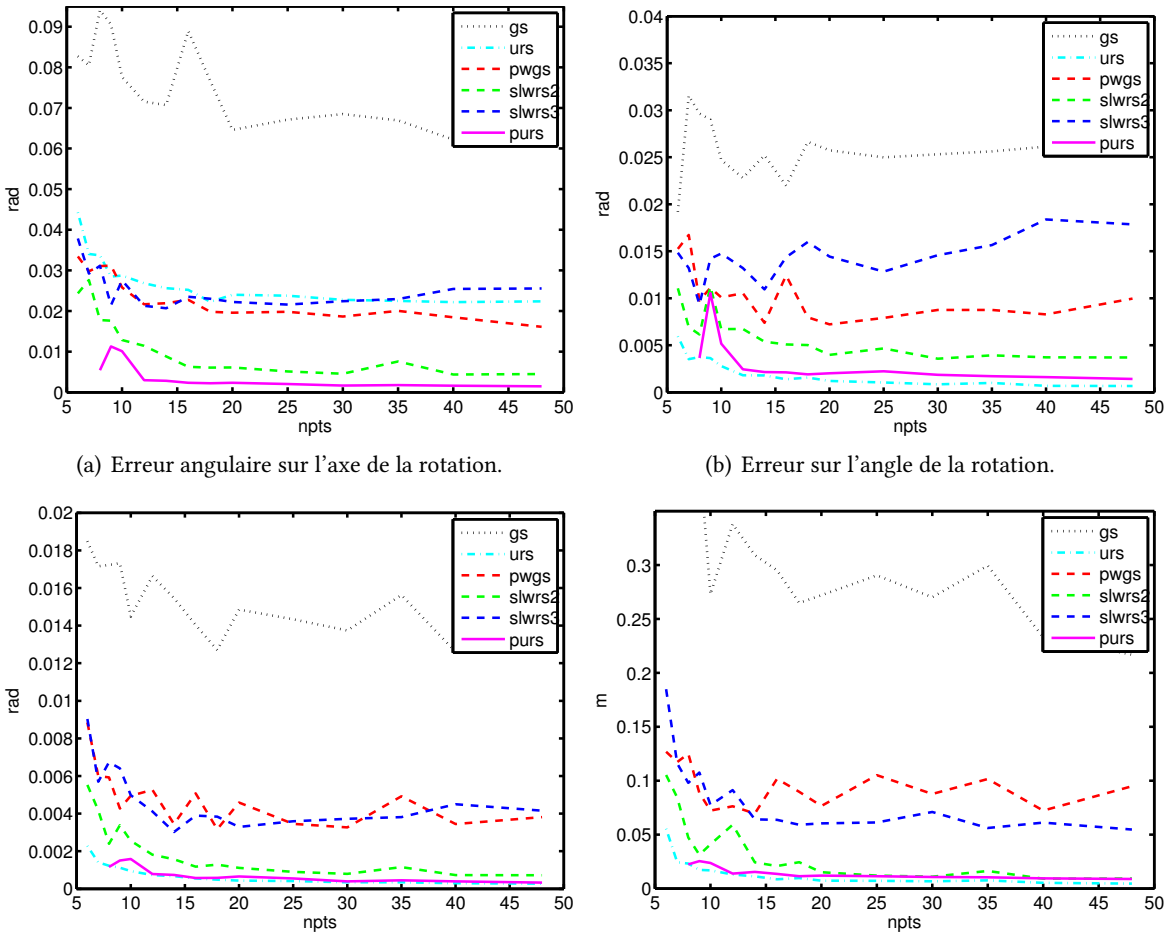
Les figures VI.4 et VI.5 présentent l'évolution de l'erreur en fonction du nombre de points utilisés pour réaliser l'estimation, respectivement dans le cas de mouvements uniformes et de mouvements quelconques. Ces expériences ont été réalisées avec un faible bruit gaussien sur les coordonnées dans l'image (de l'ordre du demi-pixel). Nous avons fait varier le nombre de points de six à quarante huit, hormis pour le modèle polynomial en raison de son instabilité lors de l'utilisation de six points. D'une manière générale, et à quelques exceptions près, le constat est toujours le même : plus nous utilisons de points et meilleure est l'estimation, tout particulièrement pour le jeu de données avec un mouvement uniforme. Les exceptions sont principalement des cas où nous utilisons une méthode prévue pour un type de mouvement donné sur des données qui sont générées avec un autre type de mouvement, par exemple :

- Le modèle «Global Shutter», conçu pour traiter des données où aucun mouvement n'est présent, présente, outre une erreur bien plus importante que les autres méthodes, une certaine constance dans l'erreur ;
- Le modèle polynomial «PURS», conçu pour traiter des données où le mouvement est uniforme, est très sujet à l'erreur lorsqu'il est utilisé avec des données issues d'un mouvement non-uniforme.

Dans le cas d'un mouvement uniforme, présenté figure VI.4, il est possible de remarquer que pour presque toutes les méthodes, de six à douze points l'erreur diminue rapidement. Au delà de quinze points, l'erreur tend lentement vers une valeur limite. Trois méthodes fournissent généralement les résultats les plus précis, il s'agît de :

- Premièrement la méthode «PURS» qui est sur les quatre paramètres étudiés la plus précise ou proche de l'être ;
- Deuxièmement la méthode «SLWRS2» qui est généralement un peu moins précise, mais jamais bien loin de la méthode précédente ;
- Troisièmement la méthode «URS» qui fait résultat égal avec la méthode «PURS» sur trois des quatre paramètres mais semble nettement moins précise sur le quatrième qui est l'axe de la rotation.

Nous constatons de toute évidence que la méthode «Global Shutter» fournit des résultats d'un ordre de précision dix fois moindre que la meilleure méthode sur chaque paramètre, ce qui est attendu étant donné que cette méthode n'est pas conçue pour des données avec un mouvement. La méthode «PWGS» donne des résultats toujours moyens. Étant donné que c'est une approximation grossière qui repose sur la méthode «Global Shutter», ce n'est pas surprenant. Elle remplit néanmoins bien son rôle d'initialisation meilleure que la méthode «Global Shutter» pour le modèle «SLWRS2». Pour terminer, la méthode «SLWRS3» donne dans l'ensemble des résultats dont la précision est du même ordre de grandeur que l'initialisation «PWGS». Comme nous le verrons dans la section suivante, cela s'explique par son ex-



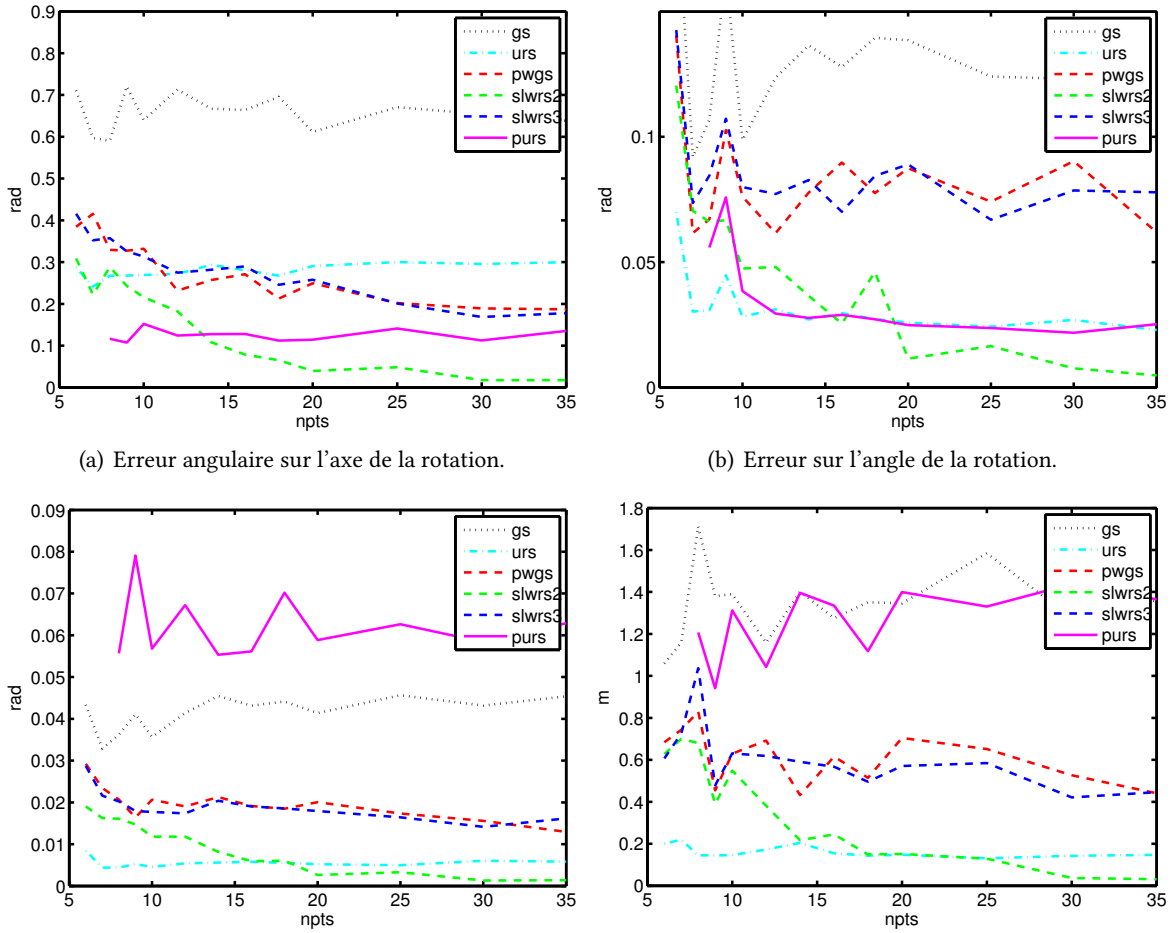
(a) Erreur angulaire sur l'axe de la rotation.

(b) Erreur sur l'angle de la rotation.

(c) Erreur angulaire sur l'axe de la translation.

(d) Erreur sur la norme de la translation.

FIGURE VI.4 – Erreur quadratique moyenne le long des lignes sur les différentes images face au nombre de points sur la séquence présentant un mouvement uniforme. Comme nous pouvons nous y attendre, d'une manière générale plus nous utilisons de points pour réaliser l'estimation, meilleur est le résultat. La méthode qui est la plus précise pour ce type de données est la méthode «PURS». Les méthodes «SLWRS2» et «URS» présentent des résultats qui ne sont pas déméritants non plus. Comme attendu, la méthode «Global Shutter» est la moins précise de toutes, mais l'approximation «PWGS» donne des résultats suffisants pour initialiser les méthodes «SLWRS». Enfin la méthode «SLWRS3» a une précision limitée, due à une trop grande sensibilité au bruit qui n'est pourtant ici que d'un demi-pixel.



(a) Erreur angulaire sur l'axe de la rotation.

(b) Erreur sur l'angle de la rotation.

(c) Erreur angulaire sur l'axe de la translation.

(d) Erreur sur la norme de la translation.

FIGURE VI.5 – Erreur moyenne face au nombre de points sur la séquence présentant un mouvement non-uniforme. À partir du moment où nous utilisons plus d'une vingtaine de points, l'estimation en utilisant la méthode «SLWRS2» est incontestablement la plus précise, ce qui s'explique par le fait qu'elle est spécialement conçue pour ce type de mouvements. Néanmoins la méthode «SLWRS3», elle aussi conçue pour ce type de données, ne fournit que des résultats moyens, dont la précision est du même ordre de grandeur que la méthode d'initialisation «PWGS». La méthode «URS» ne démérite pas, sa précision reste honorable et très proche de la méthode la plus précise sur la majorité des paramètres étudiés malgré qu'elle ait été élaborée pour des données avec un mouvement uniforme. L'hypothèse de la méthode «PURS» semble être trop grossière en ce qui concerne les paramètres de translation, lesquels sont estimés à partir des paramètres de rotation, ce qui semble poser un problème de stabilité.

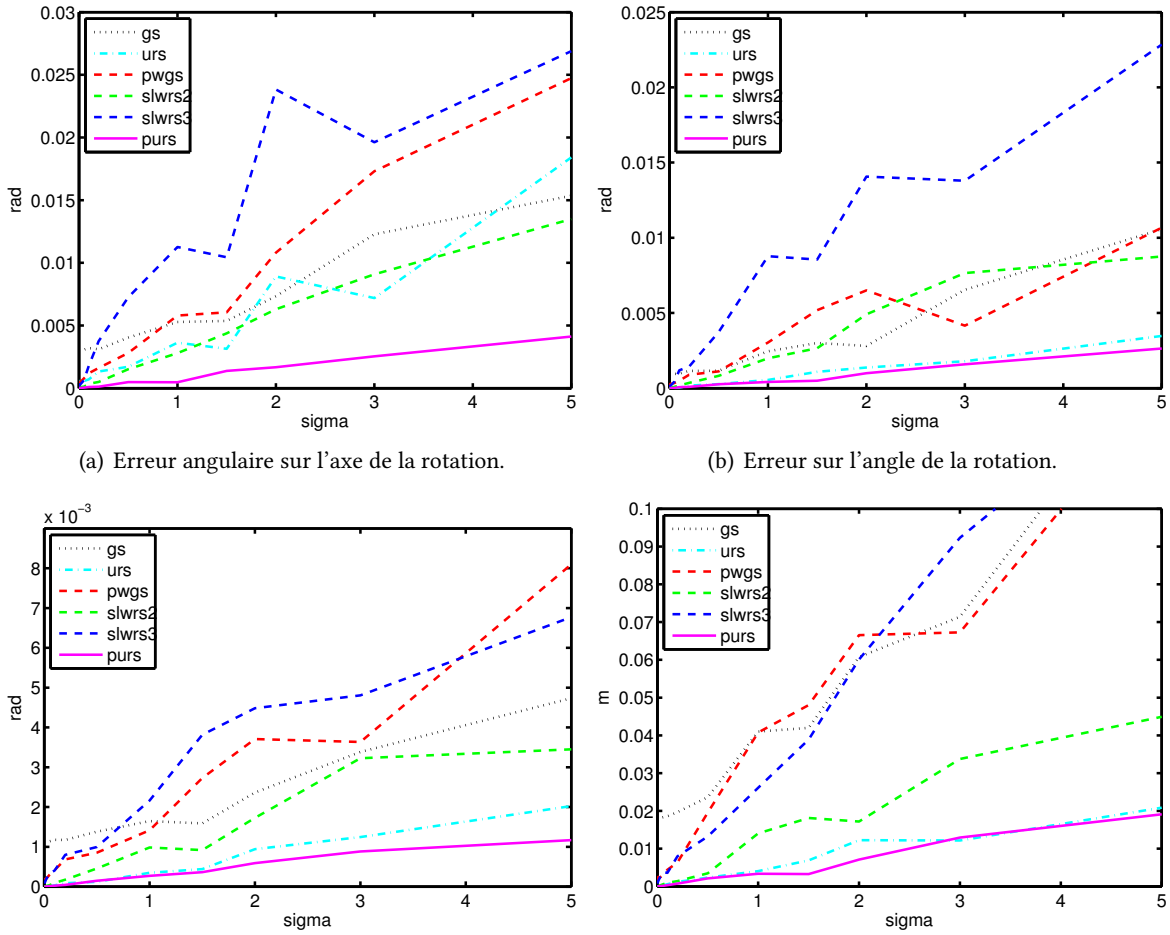
trême sensibilité au bruit.

Pour le cas du mouvement non-uniforme, la figure VI.5 nous permet de voir que la méthode la plus adaptée à ce type de données est la méthode «SLWRS2», c'est en effet elle qui donne systématiquement les résultats les plus précis, à condition de disposer d'un nombre suffisant de points (de l'ordre d'une vingtaine). Cette méthode ayant été conçue spécialement pour ce type de données, c'est un résultat attendu. La méthode «SLWRS3», elle aussi conçue pour ce type de données, donne des résultats plutôt moyens, qui sont toujours de l'ordre de deux fois plus précis que la méthode «Global Shutter», mais aussi deux à trois fois moins précis que la méthode «SLWRS2». Cela s'explique probablement là encore par sa trop grande sensibilité au bruit. Nous constatons que les résultats de la méthode «PWGS» sont du même ordre de grandeur que «SLWRS3», et remplis donc toujours bien son rôle d'initialisation, notamment pour la méthode «SLWRS2». La méthode «URS», pourtant conçue pour des données présentant un mouvement uniforme, donne des résultats étonnement bon. Elle donne en effet sur trois des quatre paramètres étudiés une erreur du même ordre de grandeur que la méthode «SLWRS2». Là encore, c'est sur l'axe de la rotation qu'elle est la moins précise. Reste la méthode «PURS», qui fournit des résultats avec une précision correcte en ce qui concerne les paramètres de la rotation, mais qui donne des résultats complètement erronés au niveau des paramètres de translation. Cela peut s'expliquer par le fait que dans cette méthode, nous optimisons d'abord les paramètres de la rotation, à partir desquels les paramètres de translation sont ensuite estimés. La matrice qui permet d'estimer les paramètres de translation à partir des paramètres de rotations est formée à l'aide de la pseudo inverse d'une matrice dont le conditionnement n'est pas toujours optimal, en particulier avec des données issues d'un mouvement non-uniforme.

VI.2.3 Comportement face au bruit

La résistance au bruit est un phénomène très étudié pour les méthodes de calcul de pose ou de mouvement à partir d'images ou de séquences vidéo. Dans le cadre des travaux présentés précédemment, cet aspect revêt une importance accrue. En effet l'estimation de mouvement à partir d'une image «Rolling Shutter» n'est possible que grâce aux déplacements des projections des points de l'objet dans l'image dûs au mouvement. Lorsque le temps d'exposition est adapté au mouvement, ces déformations peuvent être de l'ordre de plusieurs dizaines de pixels, un bruit de quelques pixels reste alors faible devant ce que nous cherchons à mesurer, d'autant que les détecteurs de points modernes comme SIFT Lowe (2004) ont une précision sub-pixelique. Néanmoins si le temps d'exposition n'est pas adapté au mouvement, les déformations sont alors au mieux de l'ordre de quelques pixels et le bruit sur la détection des points n'est plus aussi négligeable et peut donc perturber l'estimation.

Pour cette expérience, le jeu de données présentant un mouvement uniforme a été séparé en deux catégories : une avec les mouvements rapides par rapport au temps d'exposition, et donc présentant



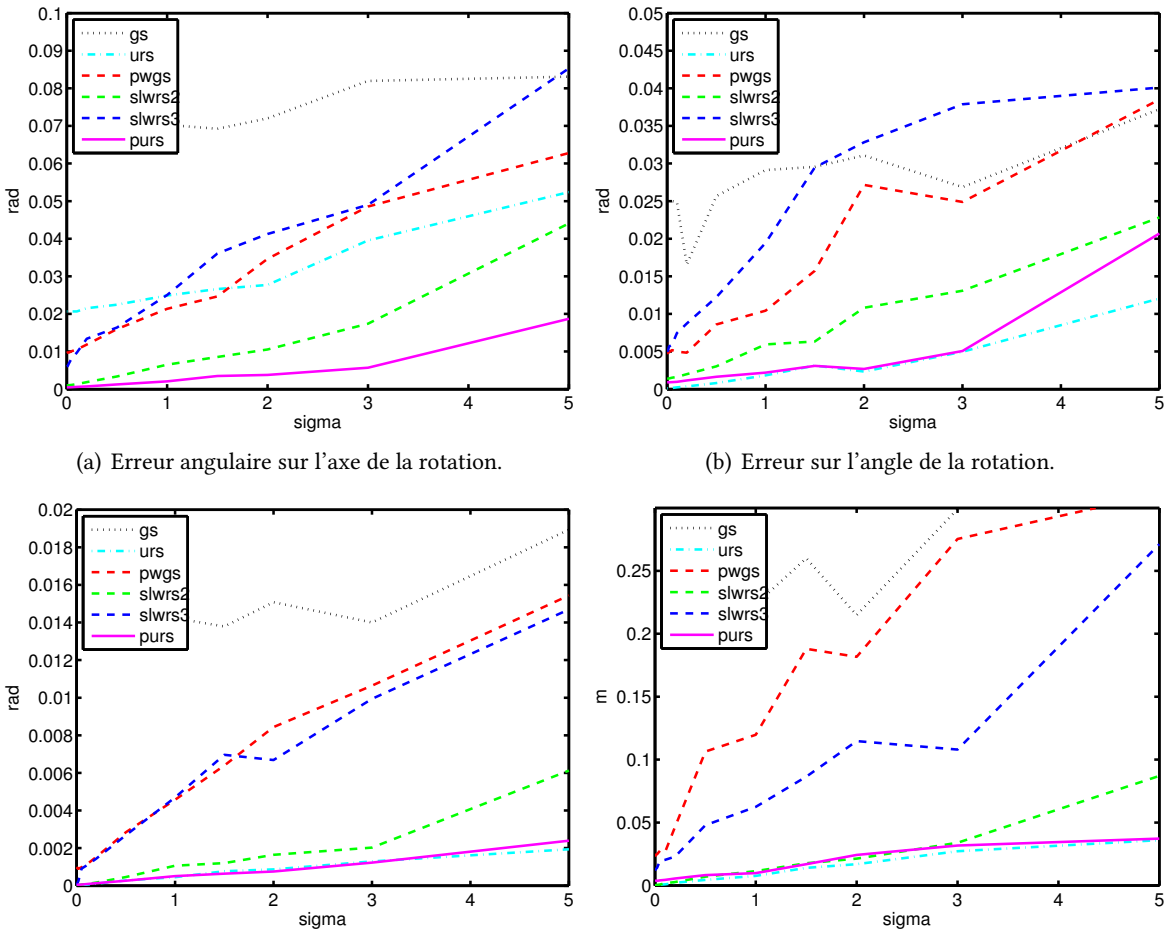
(a) Erreur angulaire sur l'axe de la rotation.

(b) Erreur sur l'angle de la rotation.

(c) Erreur angulaire sur l'axe de la translation.

(d) Erreur sur la norme de la translation.

FIGURE VI.6 – Erreur moyenne face au bruit sur la séquence présentant un mouvement uniforme rapide. Le modèle «PURS» donne les résultats les plus précis, et ce quelque soit le paramètre ou le bruit considéré. Cependant, bien que le modèle «URS» soit légèrement moins précis sur l'orientation de l'axe de la rotation, sur les autres paramètres étudiés il donne une erreur du même ordre de grandeur que le modèle «PURS». Le modèle «SLWRS2» est généralement un peu moins précis, mais fournit des résultats tout de même honorables. Les trois autres modèles («SLWRS3», «PWGS» et «GS») sont les moins précis et ont des erreurs globalement comparables.



(a) Erreur angulaire sur l'axe de la rotation.

(b) Erreur sur l'angle de la rotation.

(c) Erreur angulaire sur l'axe de la translation.

(d) Erreur sur la norme de la translation.

FIGURE VI.7 – Erreur moyenne face au bruit sur la séquence présentant un mouvement uniforme lent. Ici aussi, les modèles «PURS» et «URS» se partagent de manière serrée la meilleure précision sur trois des quatre paramètres étudiés. C'est encore sur l'orientation de l'axe de la rotation que le modèle «URS» est dépassé par le modèle «PURS». Sur cette séquence, le modèle «SLWRS2» est très proche des deux précédents : sur deux des paramètres il fournit une précision comparable. Les modèles «SLWRS3», «PWGS» et «GS» sont toujours les moins précis, notamment le modèle «GS» qui est le moins précis sur trois des quatre paramètres d'erreur.

de grandes déformations, et l'autre avec les mouvements plus lents. Les résultats correspondants sont présentés respectivement dans les figures VI.7 et VI.6. Nous remarquons immédiatement, que dans les deux cas, plus le bruit augmente et plus la précision se dégrade. C'est un phénomène habituellement observé pour les méthodes de calcul de pose. En comparant les résultats des deux catégories, nous constatons également que pour les mouvements lents, et donc présentant de faibles déformations, l'erreur moyenne est plus importante que pour les mouvements rapides. Néanmoins, bien que le calcul de pose dynamique soit donc plus robuste lorsque le temps d'exposition est judicieusement choisi, il reste dans une marge d'erreur raisonnable lorsque ce n'est pas le cas.

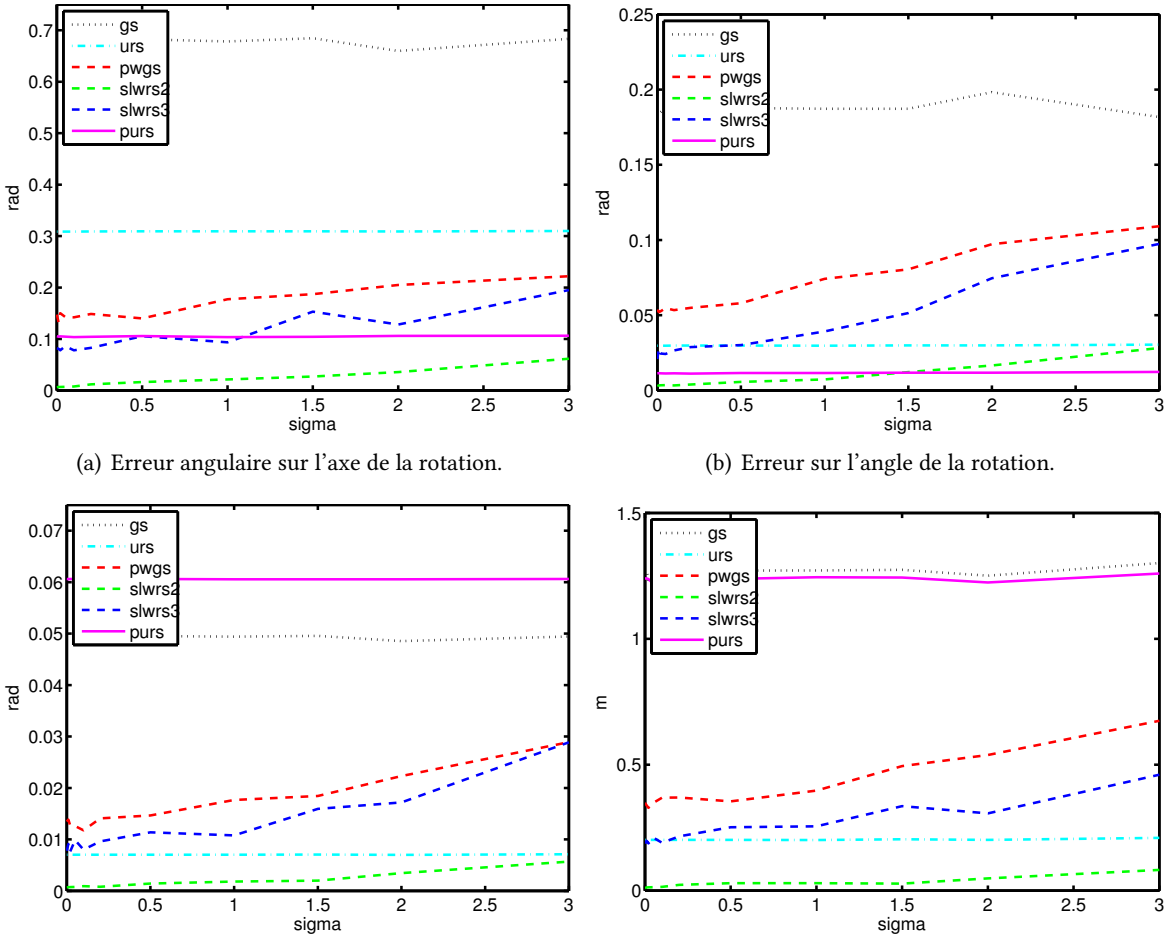
Quelque soit le mouvement uniforme, le modèle «PURS» est globalement le plus précis de tous. Lorsque le mouvement est lent, le modèle «URS» est parfois légèrement plus précis, mais ce dernier est systématiquement sujet à une erreur bien plus importante que le modèle «PURS» en ce qui concerne l'orientation de l'axe de la rotation, que le mouvement soit lent ou rapide. Cela peut être dû à la qualité de l'initialisation empirique utilisée pour cet axe, mais également à la méthode d'optimisation qui ne satisfait pas toujours parfaitement la contrainte unitaire de ce dernier. Le modèle «SLWRS2» est moins précis que les deux précédents, ce qui semble normal étant donné que ce jeu de données est uniforme, mais fournit malgré tout des résultats dont la précision reste correcte et meilleure que les trois derniers modèles («PWGS», «GS» et «SLWRS3»).

Les résultats pour la séquence à mouvement non-uniforme sont présentés dans la figure VI.8. Les modèles conçus pour les mouvements non-uniforme sont bien évidemment les plus précis, notamment le modèle «SLWRS2» qui présente les meilleurs résultats sur tous les paramètres considérés. Le modèle «SLWRS3» fournit une précision bien moindre, et comporte une erreur résiduelle même en l'absence de bruit. C'est également le cas du modèle «PWGS» qui sert d'initialisation aux deux précédents et qui présente une précision globale encore plus faible. Bien qu'il ne soit pas conçu pour ce type de mouvement, le modèle «URS» reste néanmoins relativement bon, encore une fois en dehors de l'orientation de l'axe de la rotation. La précision du modèle «PURS» reste acceptable en ce qui concerne les paramètres de rotation, en revanche pour les paramètres de translation la précision est tout au mieux du même ordre de grandeur que celle du modèle «GS», lequel est le modèle le moins précis parmi ceux présentés.

VI.2.4 Discussions

VI.2.4.1 Différences entre les modèles «PURS» et «URS»

En comparant les modèles «PURS» et «URS», il ressort deux points notables qui méritent que nous nous attardions dessus. Tout d'abord la précision sur l'estimation de l'axe de la rotation par le modèle «URS» est systématiquement moins bonne que celle par le modèle «PURS». Cela peut s'expliquer principalement pour deux raisons :



(a) Erreur angulaire sur l'axe de la rotation.

(b) Erreur sur l'angle de la rotation.

(c) Erreur angulaire sur l'axe de la translation.

(d) Erreur sur la norme de la translation.

FIGURE VI.8 – Erreur moyenne face au bruit sur la séquence présentant un mouvement non-uniforme. D'emblée nous constatons que le modèle «GS» présente l'erreur la plus élevée sur trois des quatre paramètres étudiés, dont deux où il fournit une précision nettement moindre que tous les autres modèles : les paramètres de la rotation. Bien que le modèle «PURS» a une précision comparable aux meilleurs modèles en ce qui concerne ces paramètres, il présente une erreur excessivement plus importante sur les paramètres de la translation. Les modèles «PWGS» et «SLWRS3» ont une précision globalement comparable, avec un léger gain pour le deuxième. Comme précédemment, le modèle «URS» présente une précision plutôt bonne sur trois des quatre paramètres d'erreur considérés mais celle sur l'orientation de l'axe de la rotation est assez faible. Globalement le modèle qui donne la meilleure précision pour cette expérience est incontestablement le modèle «SLWRS2».

- Lors de l'optimisation avec le modèle «URS», l'axe de la vitesse de rotation est initialisé de manière empirique, rendant le résultat final très dépendant de cette valeur, notamment si la fonction de coût présente des minima locaux aux environs du minimum global que nous recherchons, ce qui est probablement le cas ;
- La nature même de ce qui est optimisé dans les deux modèles est radicalement différente : dans le cas du modèle «URS», ce sont l'ensemble des paramètres, de translation comme de rotation, qui sont optimisés, autorisant de fait une sorte de compensation entre ces différents paramètres pour satisfaire au mieux la fonction de coût. Ainsi, localement si en s'éloignant de la solution idéalement recherchée pour les paramètres de rotation il est possible de trouver des paramètres de translation qui compensent la variation de la fonction de coût, alors cette solution peut être retenue par l'optimisation. Cela n'est pas possible avec le modèle «PURS» puisque les paramètres qui sont estimés lors de l'optimisation polynomiale sont uniquement les paramètres de rotation. Plus exactement l'élimination des paramètres de translation lors de la construction de la fonction de coût polynomiale peut s'interpréter comme le fait de chercher les paramètres de rotation tel qu'il est impossible d'expliquer autrement les données que par la rotation.

Ensuite nous avons remarqué que l'erreur sur les paramètres de translation estimés avec le modèle «PURS» sur des données présentant un mouvement non-uniforme est bien plus importante que celle obtenue avec le modèle «URS». Là encore ceci s'explique par le fait que lors de l'optimisation polynomiale nous estimons uniquement les paramètres de rotation avec le modèle «PURS». Nous laissons ainsi aux paramètres de translation le soin d'expliquer au mieux les incohérences dans les données résultant de l'inadéquation avec l'hypothèse uniforme, ce qui provoque de fait une erreur plus importante sur ces derniers.

VI.2.4.2 Réglage du temps d'exposition

Dans les travaux présentés, nous avons toujours considéré le temps d'exposition d'une ligne comme une valeur établie à l'avance sans plus expliquer son réglage. Cependant les résultats sur les données simulées obtenus en séparant celles-ci en mouvement lent et rapide fait apparaître combien ce réglage est important. Celui-ci doit se faire avec un principe simple en tête : plus les déformations dues au mouvement sont marquées dans l'image et meilleure est la précision sur l'estimation des paramètres du mouvement. Cela signifie qu'il faut prendre en compte lors de ce réglage le mouvement attendu de l'objet selon trois critères :

- La nature même du mouvement car une rotation engendre des déformations plus importantes qu'une translation par exemple ;
- L'axe principal du mouvement puisque, par exemple, une translation le long de l'axe optique provoque une déformation moindre que le long d'un des axes du capteur ;

- Et enfin la vitesse probable du mouvement relativement à la profondeur de celui-ci, en effet un mouvement à grande vitesse filmé de très loin implique des déformations moins visible que si il était filmé plus proche, et à distance égale un mouvement rapide engendre plus de déformations qu'un mouvement lent.

VI.2.5 Mise en correspondance automatique

Nous avons vu section V.4 que la méthode «PURS» permettait de réaliser une mise en correspondance automatique entre les points du modèle tridimensionnel et leurs projections dans l'image. Nous nous attachons ici à tester la robustesse de l'estimation en présence de correspondances erronées en utilisant le jeu de données simulées présentant un mouvement uniforme et un bruit de l'ordre du quart de pixel sur les coordonnées des 40 points dans l'image. Dans une application réelle, les correspondances erronées peuvent être de deux types :

- Un point du modèle tridimensionnel est mis en correspondance avec la projection d'un autre point du modèle dans l'image ou deux correspondances sont inversées, c'est fréquent lorsque la texture de l'objet est répétitive ;
- Un point du modèle tridimensionnel est mis en correspondance avec un point aberrant de l'image (le fond, un autre objet de la scène, ...).

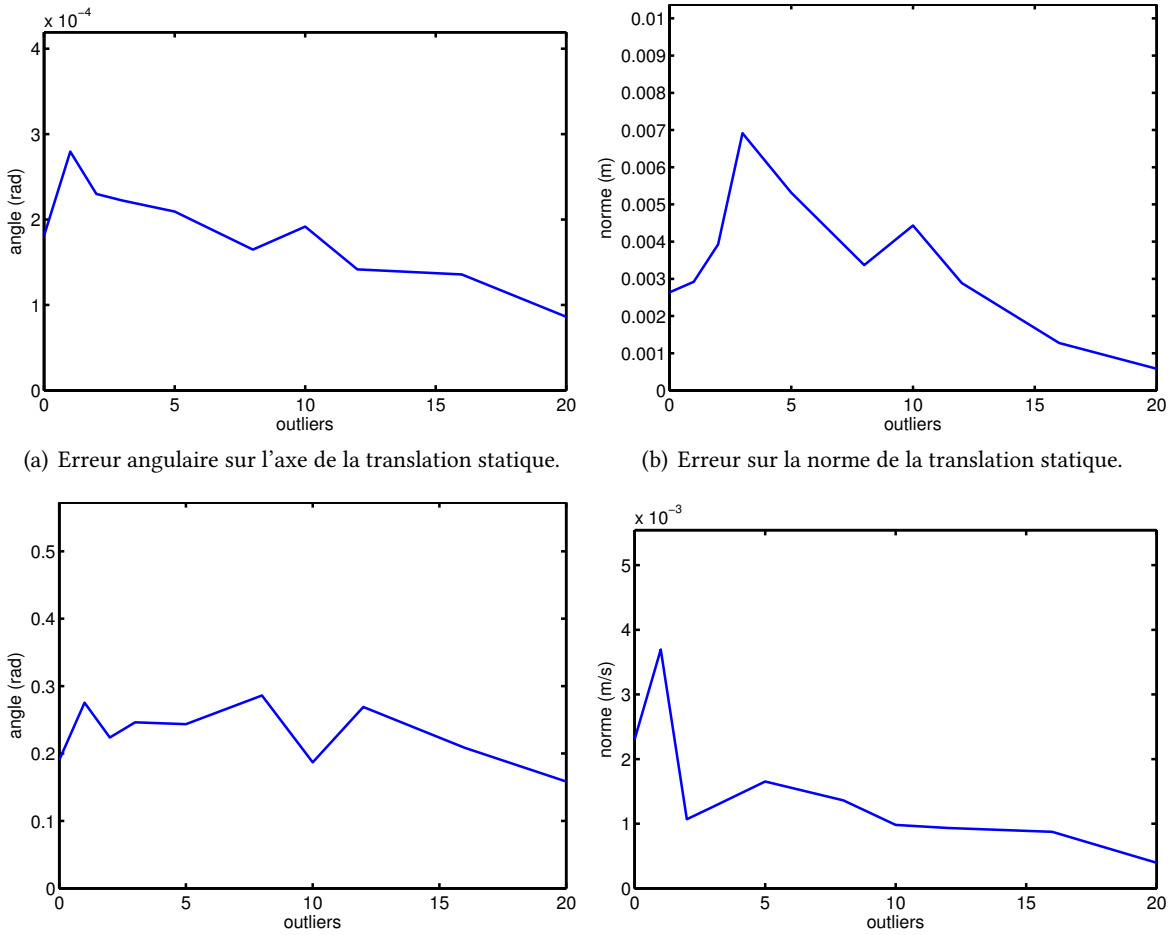
Dans notre expérience nous avons simulé ces deux types d'erreurs de la manière suivante :

- En échangeant les coordonnées de deux correspondances ou en substituant les coordonnées d'une correspondance par celles d'un autre point ;
- En remplaçant les coordonnées d'une correspondance par des coordonnées aléatoires dans l'image.

Leur nombre total varie de 0 à 20, ce qui correspond à un taux de 0 à 50%, et nous nous intéressons alors à l'évolution du bruit. Les résultats de cette expérience sont présentés dans les figures VI.9 et VI.10. Ces différentes courbes sont relativement constantes et peu perturbées par l'évolution du nombre de correspondances erronées. Cela démontre bien l'utilité de «RANSAC» pour rendre robuste l'estimation des paramètres.

Cette méthode devant faire le tri entre les correspondances correctes et celles erronées, il est également intéressant de s'intéresser au nombre d'erreurs lors de ce tri. Habituellement ces erreurs peuvent être séparées en deux classes :

- Les faux-positifs, c'est à dire les correspondances erronées qui ont été considérées comme correctes. Elles sont de possibles sources d'erreurs lors de l'estimation des paramètres, ou d'imprécisions conséquentes. En réglant correctement les seuils de RANSAC, nous sommes parvenus à éliminer presque totalement ce type d'erreurs.
- Les faux-négatifs, c'est à dire les correspondances correctes qui ont été considérées comme erronées. De fait elles ne participent pas à l'estimation des paramètres et diminuent donc la qualité



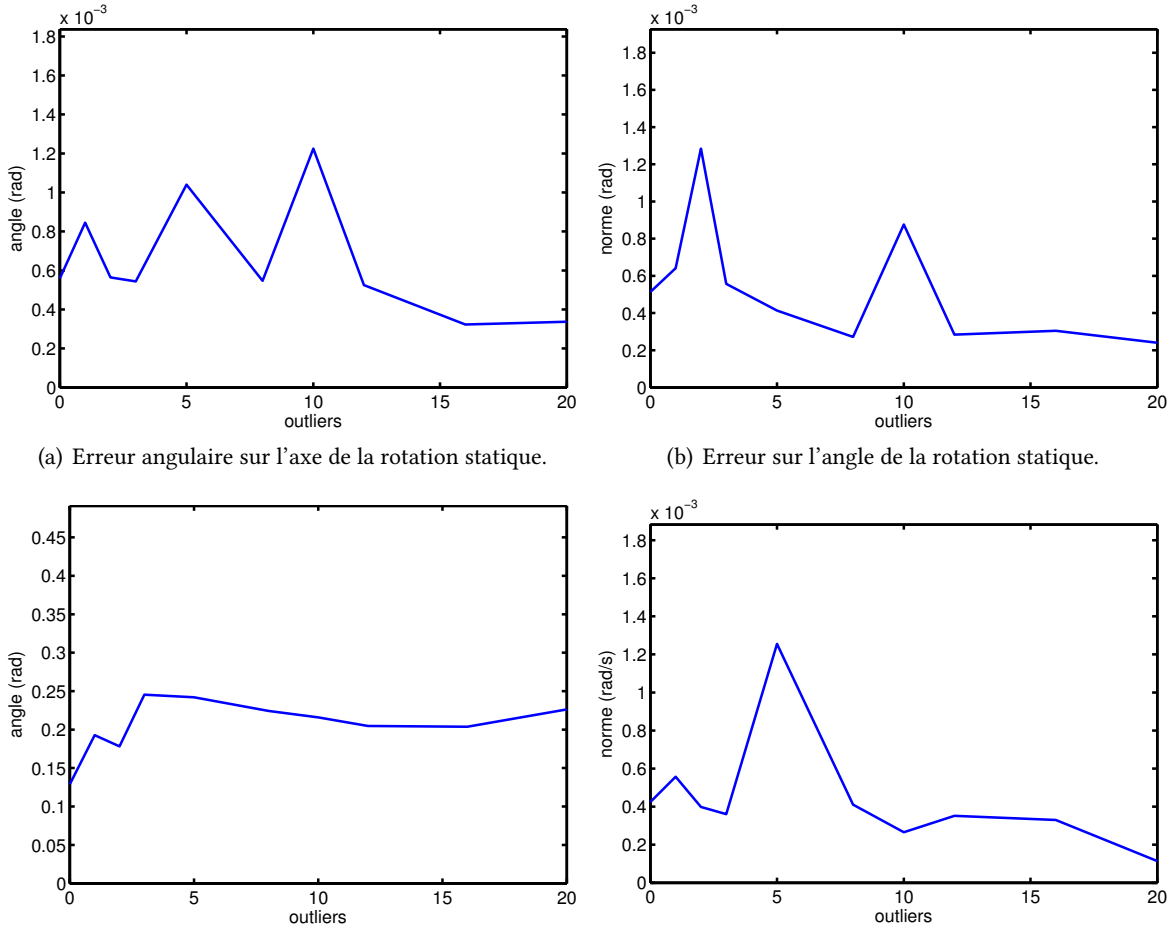
(a) Erreur angulaire sur l'axe de la translation statique.

(b) Erreur sur la norme de la translation statique.

(c) Erreur angulaire sur l'axe de la vitesse de translation.

(d) Erreur sur la norme de la vitesse de translation.

FIGURE VI.9 – Erreurs moyennes sur les paramètres de translation par rapport au nombre de correspondances erronées. Celles-ci sont globalement constantes et peu perturbées par le nombre de correspondances erronées.



(a) Erreur angulaire sur l'axe de la rotation statique.

(b) Erreur sur l'angle de la rotation statique.

(c) Erreur angulaire sur l'axe de la rotation dynamique.

(d) Erreur sur la norme de la vitesse de rotation.

FIGURE VI.10 – Erreurs moyennes sur les paramètres de rotation par rapport au nombre de correspondances erronées. Celles-ci sont globalement constantes et peu perturbées par le nombre de correspondances erronées.

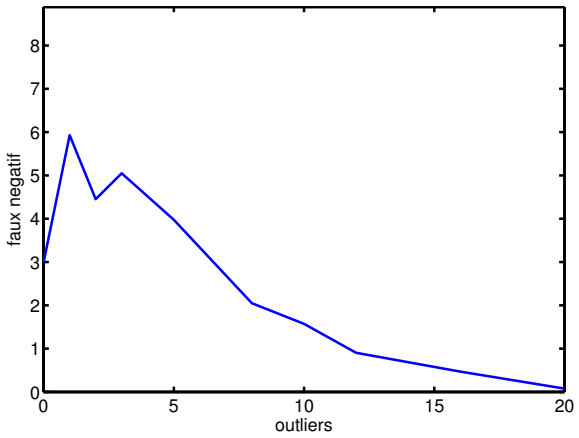


FIGURE VI.11 – Évolution du nombre de faux-négatifs en fonction du nombre de correspondances erronées. Plus il y a de correspondances erronées dans les données, moins il y a de potentiels faux-négatifs.

de l'estimation. La figure VI.11 présente l'évolution du nombre de faux-négatifs en fonction du nombre de correspondances erronées.

VI.3 Résultats avec des données réelles

Deux types d'expériences sur des mouvements d'objets réels ont été effectuées. Dans un premier temps un objet non texturé a été filmé sur deux mouvements simples : une rotation plane et une translation rectiligne. Les correspondances avec le modèle tridimensionnel ont été réalisées à la main. Dans un deuxième temps, des objets texturés ont été utilisés avec divers mouvements : chute libre, rotation et translation plane. La méthode décrite section V.4 a été utilisée pour réaliser les correspondances.

VI.3.1 Avec correspondances connues

Il faut noter que dans cette section seules les méthodes «URS» et «PURS» sont présentées en raison du faible nombre de points mis en correspondances dans les séquences utilisées, et ne permettant donc pas d'appliquer le modèle «SLWRS» dans la plupart des images.

VI.3.1.1 Séquence en translation

Le première séquence, qui est présentée partiellement figure VI.12, montre un objet filmé au cours d'une translation le long d'un rail rectiligne. La séquence complète contient dix images. L'objet est constitué de deux faces planes contenant seize marqueurs circulaires blancs et formant un angle droit. Dans la scène, l'objet se déplace de l'arrière vers l'avant et du coin supérieur gauche au coin inférieur droit.

Les déformations correspondantes au mouvement d'arrière en avant sont difficiles à observer, néanmoins sur l'image où les déformations sont le plus prononcées, nous pouvons mesurer la distance horizontale entre les marqueurs qui sont sur la face verticale et observer qu'elle augmente de haut en bas. En revanche les déformations correspondantes au mouvement le long des axes horizontal et vertical du capteur sont clairement visibles. L'objet est étiré de haut en bas et subit l'effet «penché», effet qui est décrit section III.2.

La vitesse et la position estimées de l'objet par les méthodes «PURS» et «URS» tout au long de cette séquence sont données dans la figure VI.13. Nous pouvons remarquer d'emblée que les résultats des deux méthodes sont très semblables, ce qui est attendu étant donné les résultats obtenus dans la section VI.2. Le mouvement est globalement conforme à celui d'un objet en translation rectiligne. Pour les trois premières images, la vitesse en translation n'est pas parfaitement rectiligne, de même que la vitesse de rotation est non nulle bien que très faible. Cela est probablement dû au fait que l'objet n'était pas parfaitement collé au rail au tout début du mouvement. La vitesse en rotation est ensuite nulle sur le reste de la séquence, et la translation est parfaitement rectiligne jusqu'aux deux dernières images, lorsque l'objet termine sa course. Ceci s'explique par le relâchement de la pression sur l'objet.

(a) (b)

(c) (d)

(e) (f)

FIGURE VI.12 – Images extraites de la séquence de mouvement réel en translation rectiligne avec un objet tridimensionnel. Nous observons des déformations principalement caractéristiques d'un mouvement le long des axes horizontal (effet «penché») et vertical (élongation) du capteur. Les déformations résultantes de la translation le long de l'axe optique sont indiscernables sans mesure.

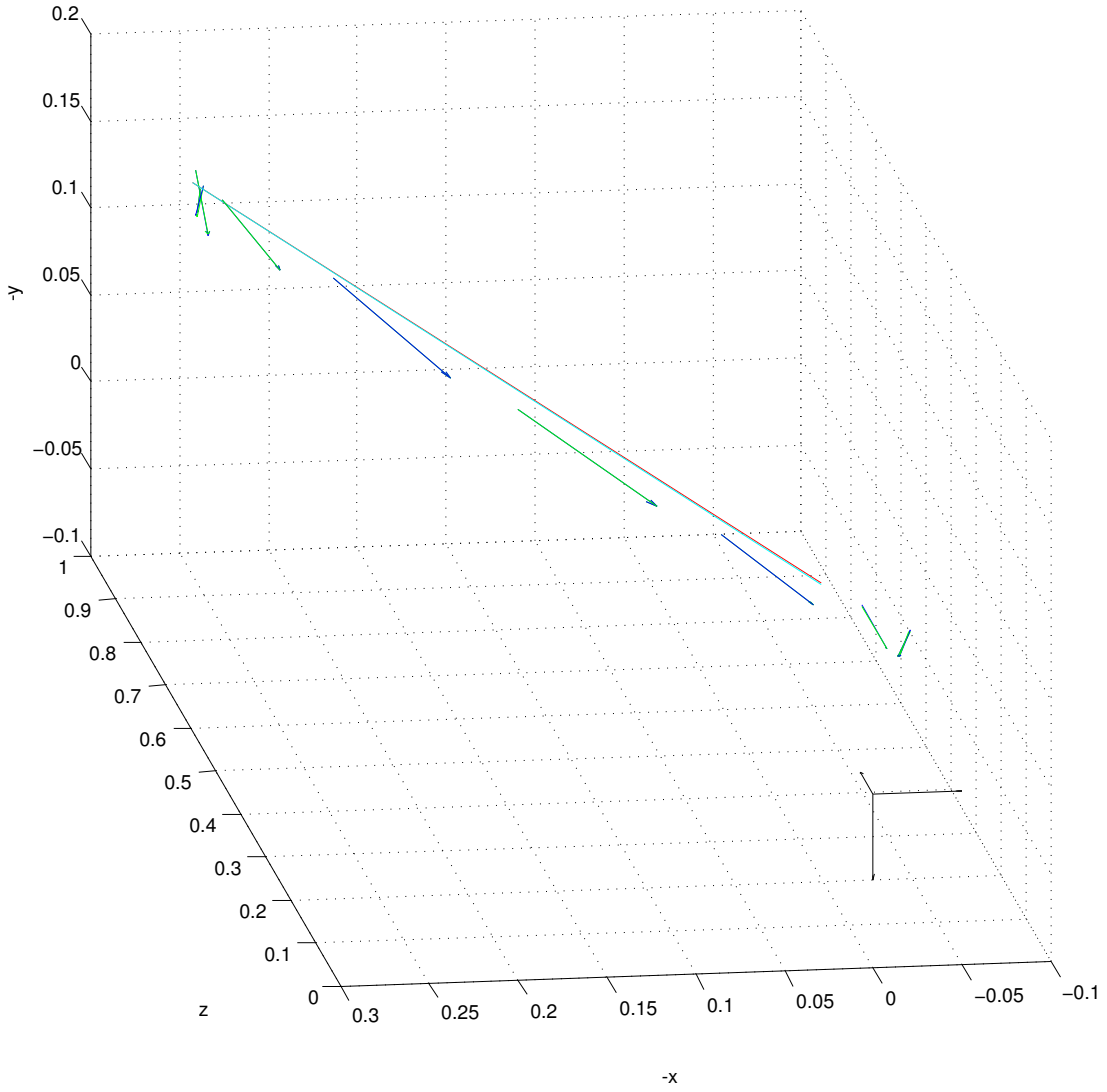


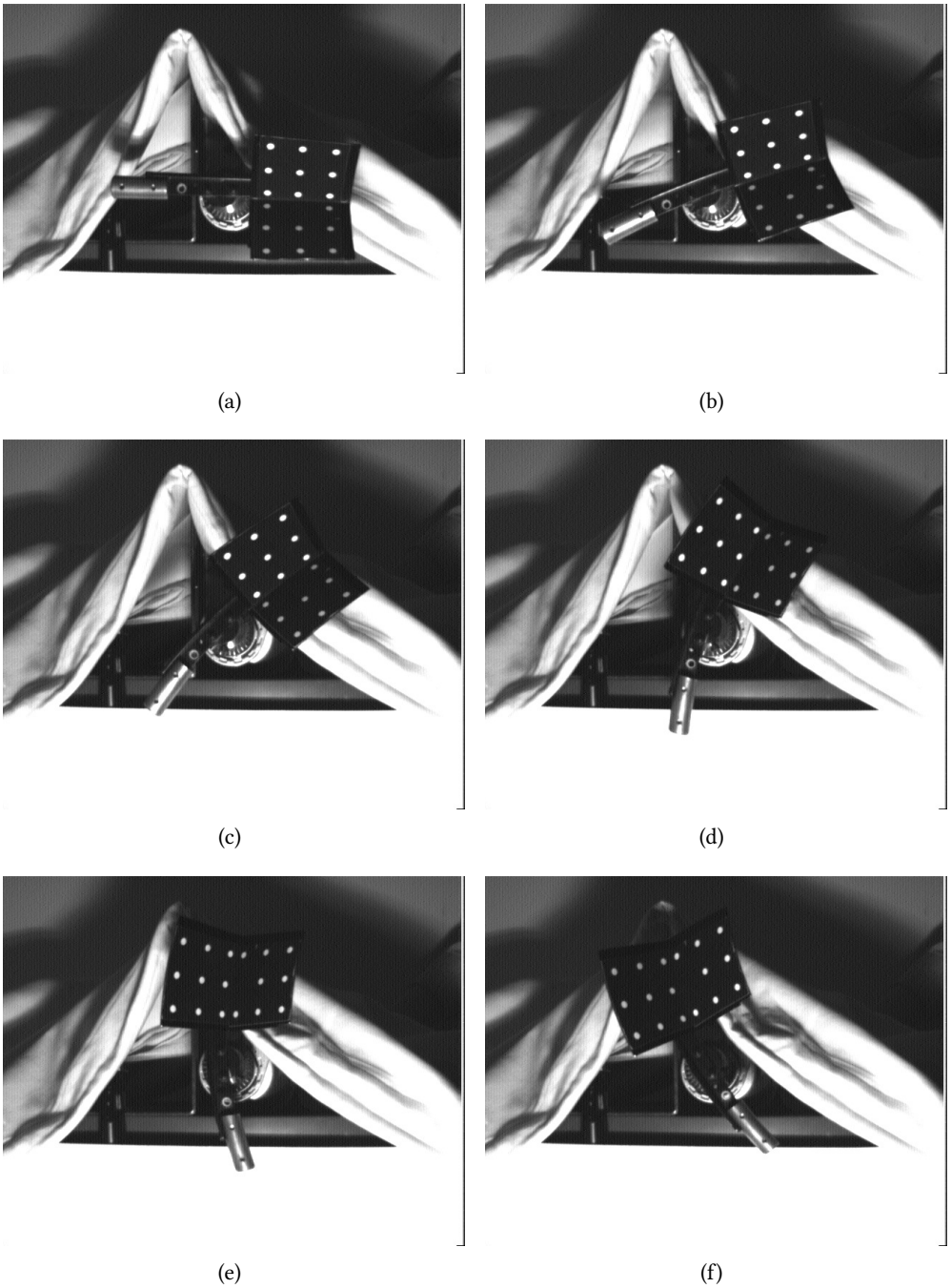
FIGURE VI.13 – Les flèches vertes indiquent la position et la vitesse de l'objet estimées avec le modèle «PURS». Les flèches bleues correspondent à celles estimées avec le modèle «URS». La droite cyan est la droite moyenne passant par les positions de l'objet estimées avec le modèle «PURS». La droite rouge est celle passant par les positions de l'objet estimées avec le modèle «URS». Le repère noir indique la position de la caméra. Le mouvement estimé reflète bien le mouvement réel de l'objet.

VI.3.1.2 Séquence en rotation

La deuxième séquence, qui est présentée partiellement figure VI.14, montre un objet filmé au cours d'une rotation plane quasiment fronto-parallèle. La séquence complète comporte vingt-huit images. L'objet est constitué de deux faces planes contenant dix-sept marqueurs circulaires blancs et formant un angle droit. Il est monté sur un support qui est fixé au rotor d'une perceuse. L'objet réalise presque deux tours autour de l'axe de rotation. Au cours des seize premières images, qui constituent le premier tour, il accélère légèrement tandis qu'au deuxième tour, correspondant aux douze images suivantes, sa vitesse est relativement stable.

Nous pouvons observer dans l'image des déformations typiques de la rotation autour de l'axe optique de la caméra. En effet, le support de l'objet, qui est droit dans la réalité, se retrouve courbé dans la majorité des images. Il en va de même pour les bords des plaques qui constituent l'objet. Ce dernier apparaît par ailleurs compressé dans la première image de la figure VI.14. Sur les images, non visibles ici, où l'objet est dans l'autre moitié de l'image, il est étiré. Ce sont bien des effets qui ont été décrits section III.2.

Le mouvement estimé au premier tour est présenté dans la figure VI.15 tandis que la figure VI.16 montre celui estimé lors du deuxième tour. Durant le début du premier tour nous retrouvons bien l'accélération progressive de l'objet. Lors du deuxième tour, nous avons bien une rotation s'effectuant à vitesse quasi constante. Cela est encore plus visible dans la figure VI.17 qui présente le mouvement estimé projeté dans le plan moyen de la rotation. Nous pouvons cependant remarquer que l'axe de la rotation, le centre du cercle, est légèrement décalé au premier tour et plus franchement au second. Les deux modèles comparés, «URS» et «PURS», fournissent néanmoins des résultats très proches. Étant donné que le mouvement est relativement simple et que la vitesse reste raisonnable, cela est normal à la vue des résultats de la section VI.2.



(a) (b)

(c) (d)

(e) (f)

FIGURE VI.14 – Images extraites de la séquence de mouvement réel en rotation plane avec un objet non texturé. Les lignes droites, comme le bord de l'objet et le montant, sont courbées. C'est un effet caractéristique des rotations.

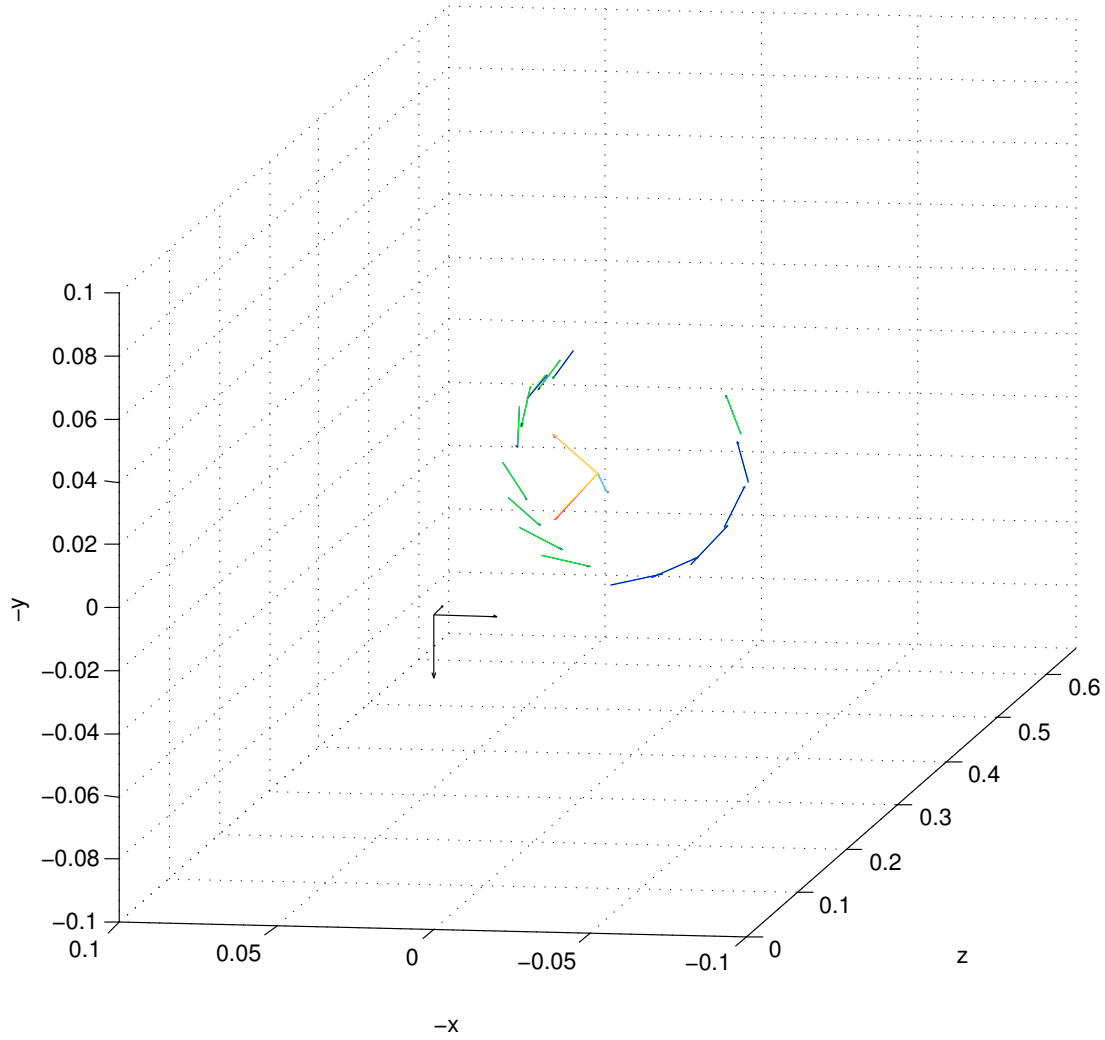


FIGURE VI.15 – Les flèches vertes indiquent la position et la vitesse estimées à l'aide du modèle «PURS». Celles estimées avec le modèle «URS» sont indiquées par des flèches bleues. La flèche cyan est un vecteur normal au plan moyen formé par les positions de l'objet estimées avec le modèle «PURS», et les flèches jaunes forment une base de ce plan. La flèche rouge (confondue avec celle cyan) et les flèches de couleur magenta correspondent à l'estimation à l'aide du modèle «URS». Le repère noir indique l'emplacement de la caméra. Nous retrouvons bien le mouvement de rotation de l'objet lors du premier tour. Il accélère progressivement lors des premières images.

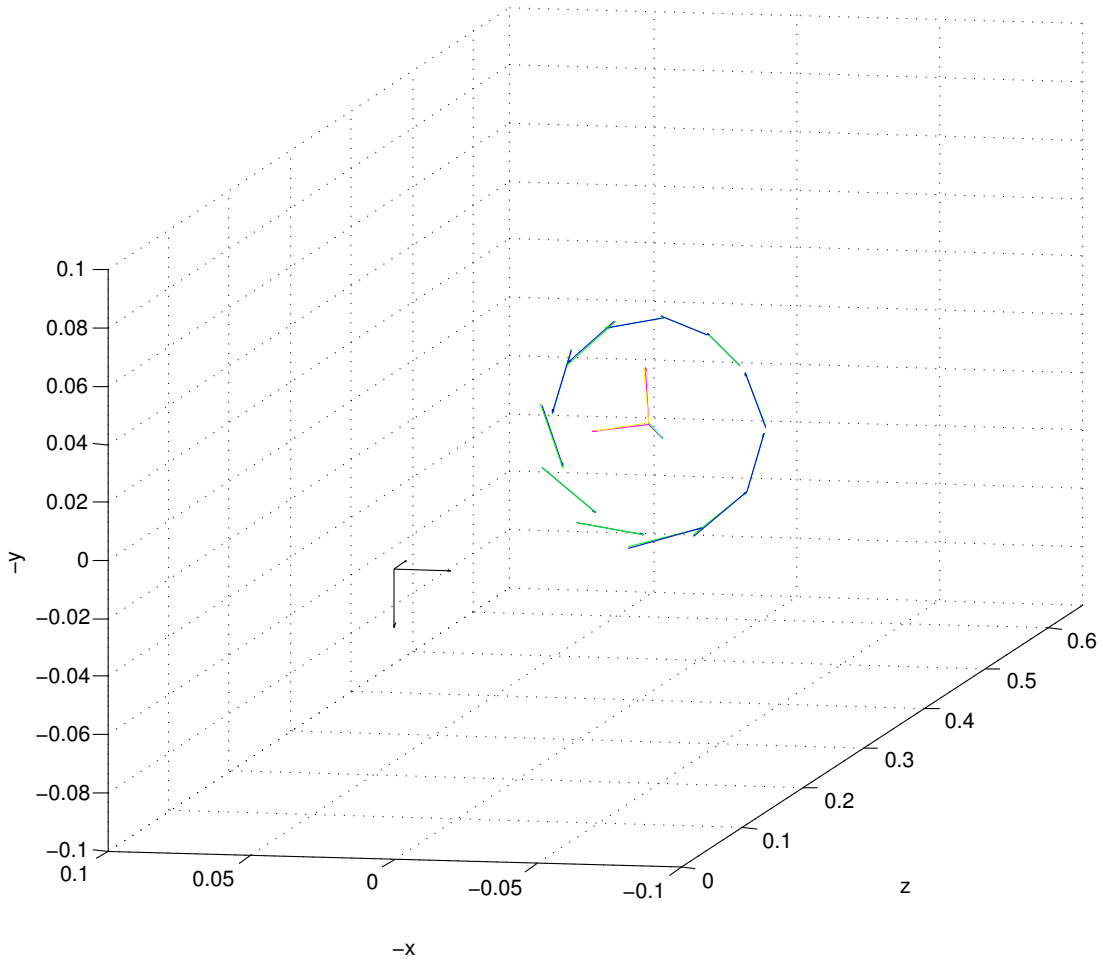
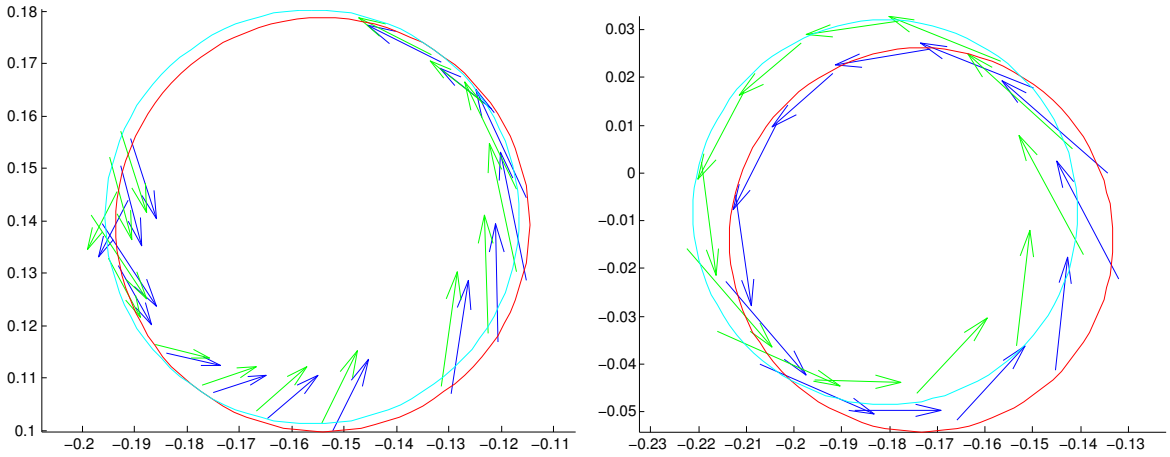


FIGURE VI.16 – Les flèches vertes indiquent la position et la vitesse estimées à l'aide du modèle «PURS». Celles estimées avec le modèle «URS» sont indiquées par des flèches bleues. La flèche cyan est un vecteur normal au plan moyen formé par les positions de l'objet estimées avec le modèle «PURS», et les flèches jaunes forment une base de ce plan. La flèche rouge (confondue avec celle cyan) et les flèches de couleur magenta correspondent à l'estimation à l'aide du modèle «URS». Le repère noir indique l'emplacement de la caméra. Nous retrouvons bien le mouvement de rotation de l'objet lors du deuxième tour. La vitesse semble bien constante.



(a) Premier tour (b) Deuxième tour

FIGURE VI.17 – Le mouvement estimé est projeté sur le plan moyen passant par les positions de l'objet estimées avec chacune des méthodes. Les flèches vertes donnent la position et la vitesse estimées avec le modèle «PURS» tandis que celles bleues correspondent au modèle «URS». Pour chaque modèle un cercle est ajusté aux positions estimées, il est cyan pour le modèle «PURS» et rouge pour le modèle «URS». Les cercles sont presque confondus lors du premier tour, et on remarque bien que l'objet accélère tandis qu'au deuxième tour la vitesse est plus constante, mais les cercles sont légèrement plus décalé.

(a) Vue caméra gauche (b) Vue caméra droite

FIGURE VI.18 – Vues d'une caméra stéréoscopique observant la boite utilisée comme objet dans l'expérience de la chute libre.

VI.3.2 Mise en correspondance automatique

Dans cette section, nous nous intéressons à la mise en correspondance automatique sur des objets texturés en suivant la méthode décrite section V.4. Dans ce but deux expériences ont été réalisées avec des objets tridimensionnels texturés. La première fait apparaître, sur une seule image, une boite en chute libre. La deuxième est une séquence de dix-huit images où une tasse est en mouvement sur une table.

VI.3.2.1 Chute libre d'une boite

La méthode a tout d'abord été testée sur une image présentant un objet tridimensionnel texturé qui subit une chute libre. L'objet en question est une boite sur laquelle ce trouve une illustration comportant de nombreux détails. La figure VI.18 montre une vue stéréoscopique statique de celle-ci obtenue à partir d'une paire de caméras stéréoscopique calibrée. Le modèle tridimensionnel obtenu par reconstruction est présenté dans la figure VI.19, en vue de dessus. Nous retrouvons bien les deux faces de la boite ainsi que le coin arrondi entre les deux. La reconstruction contient un peu plus de trois-cents points, dont une vingtaine qui sont des points erronés.

Dans la figure VI.20 nous présentons l'image obtenue en faisant chuter librement cette boite devant une caméra «Rolling Shutter». La figure VI.21 montre la position et le mouvement estimés de la boite dans le repère caméra. Environ deux-cents points ont alors été mis en correspondance avec les points du modèle tridimensionnel grâce à «SIFT» (Lowe, 2004). Après le filtrage par «RANSAC», une cinquantaine de points ont été conservés pour l'estimation du mouvement. Malgré la présence de trois faux-positifs

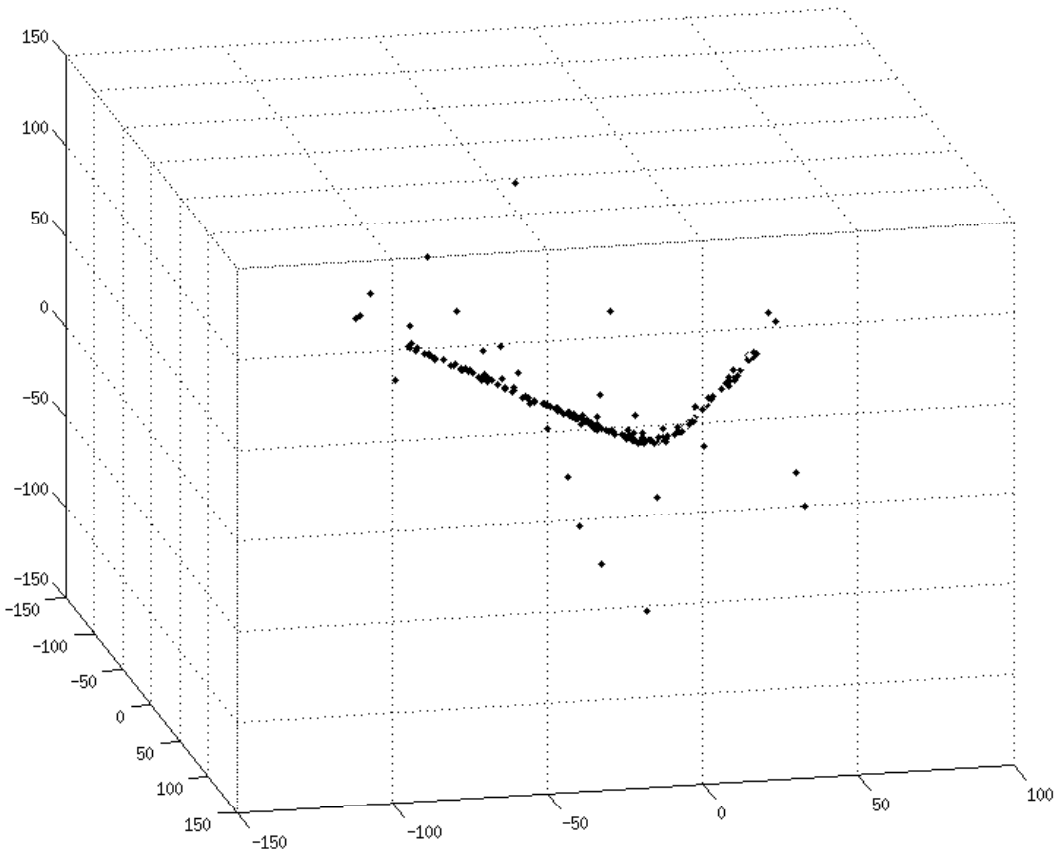


FIGURE VI.19 – La reconstruction tridimensionnelle de la boite utilisée comme objet pour l'expérience de la chute libre contient un peu plus de trois-cents points, dont une vingtaine sont de toute évidence erronés. La boite est vue de dessus afin de mieux la reconnaître.

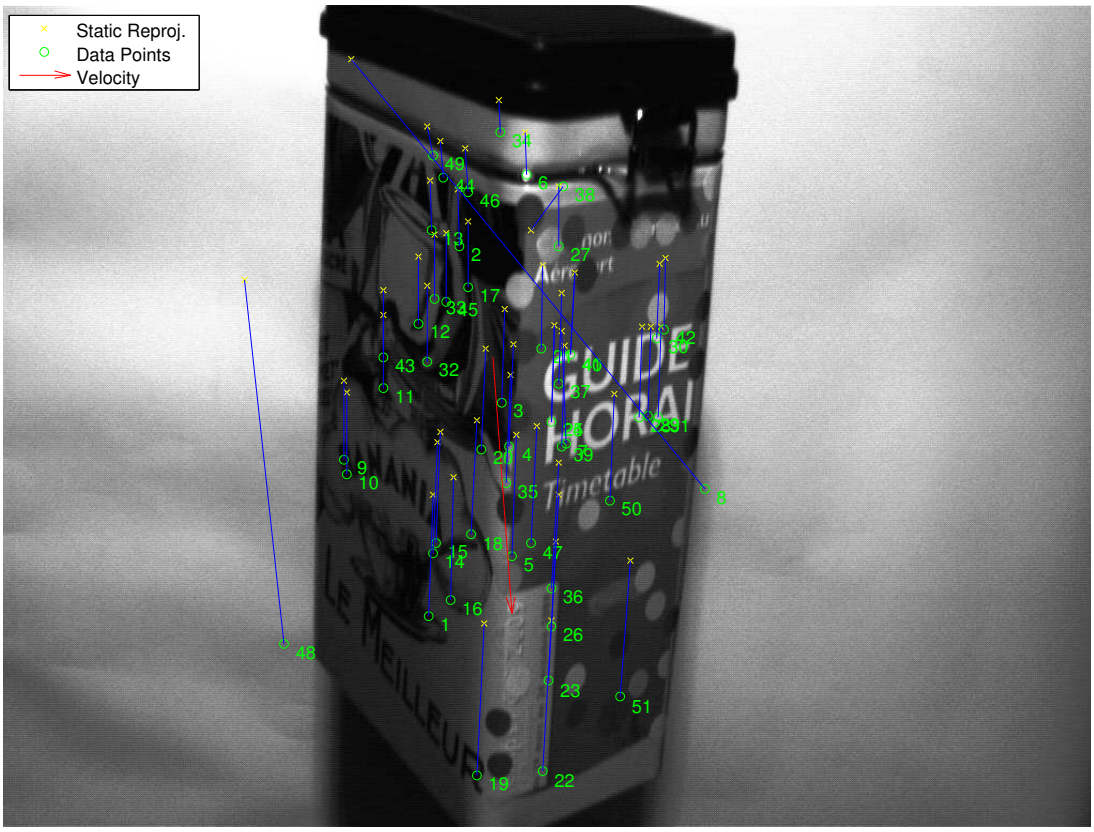


FIGURE VI.20 – Image obtenue en laissant chuter la boite devant une caméra «Rolling Shutter». Les points verts sont les points mis en correspondances avec le modèle tridimensionnel. Les points jaunes indiquent la reprojection avec les paramètres statiques de la pose estimée. Le mouvement estimé de l'objet est donné par la flèche rouge. Cette estimation est cohérente avec le mouvement réel de l'objet et les conditions de la prise de vue.

dans ces points, nous retrouvons bien le mouvement de chute de l'objet vers le bas, accompagné d'une très faible vitesse de rotation, l'objet n'ayant pas été lâché parfaitement droit. Les faux-positifs sont le résultat de la difficulté à régler correctement les différents seuils de «RANSAC» pour cette séquence. De plus, étant donné la faible proportion de points en correspondance correcte, le nombre d'itérations de «RANSAC» nécessaire pour obtenir ce résultat a été particulièrement élevé (plusieurs dizaines de milliers).

VI.3.2.2 Tasse en mouvement sur une table

La séquence suivante fait apparaître une tasse en mouvement plan sur une table. La tasse est présentée dans la vue stéréoscopique de la figure VI.22. Elle est hautement texturée et le modèle tridimensionnel est reconstruit à partir de deux-cents vingts points. Il est présenté dans la figure VI.23 et il

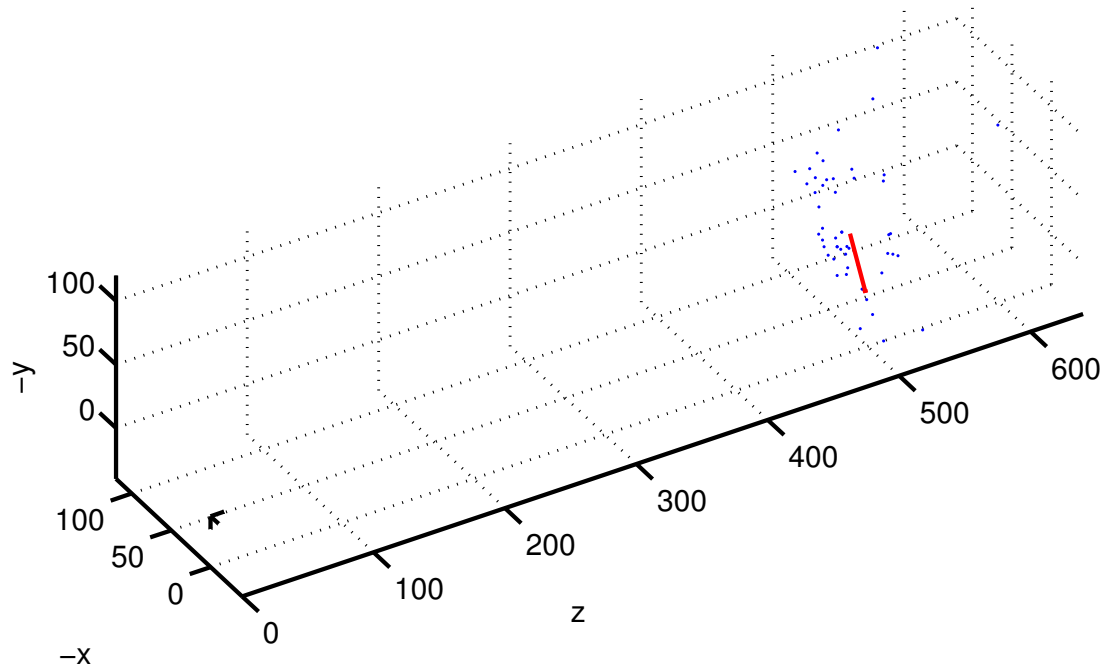


FIGURE VI.21 – Position de la boite dans le repère caméra et son mouvement estimés lors de la chute libre. La flèche rouge indique la vitesse de translation. Celle-ci reflète bien la chute libre de la boite étant donné que la caméra était positionnée avec une vue plongeante sur la boite.

(a) Vue caméra gauche (b) Vue caméra droite

FIGURE VI.22 – Vues d'une caméra stéréoscopique observant la tasse utilisée comme objet dans l'expérience de la table.

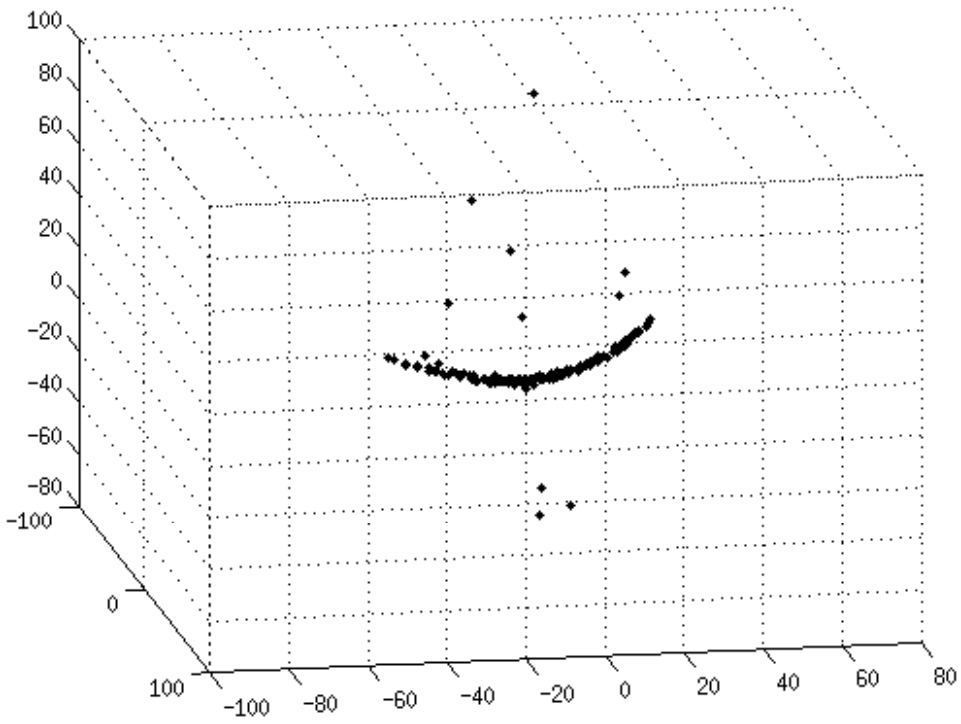


FIGURE VI.23 – La reconstruction tridimensionnelle de la tasse utilisée comme objet pour l'expérience de la table. Le modèle contient un peu plus de deux-cents vingts points, dont une douzaine sont de toute évidence erronés. Nous retrouvons bien la forme de la tasse, ici vue sur le profilé de celle-ci.

contient une douzaine de points erronés. La séquence complète contient dix-huit images et comporte quatre mouvements différents. Les mouvements sont une combinaison de translation sur le plan de la table et de rotations d'axe normal au plan.

La figure VI.24 présente les résultats obtenus sur une des images de la séquence. La séquence complète se trouve sur https://vimeo.com/106113458. Le mouvement estimé est globalement conforme au mouvement réel :

- Les axes de rotation sont tous relativement parallèles, et normaux au plan de la table.
- Les axes de translation sont également tous relativement parallèles, les translations sur le plan de la table étant globalement rectilignes.

Sur quelques unes des images il est possible de noter la présence de faux-positifs qui ont perturbés l'estimation. Les seuils de «RANSAC» ont pourtant été réglés de manière assez stricte de sorte à limiter grandement le nombre de faux positifs dans cette séquence. Malgré ce réglage très strict, et en raison

de la bonne qualité de la texture de l'objet, dans cette séquence le taux de correspondances correctes variait de deux tiers à trois quarts, ce qui a limité le nombre d'itérations nécessaires pour «RANSAC» à quelques centaines.

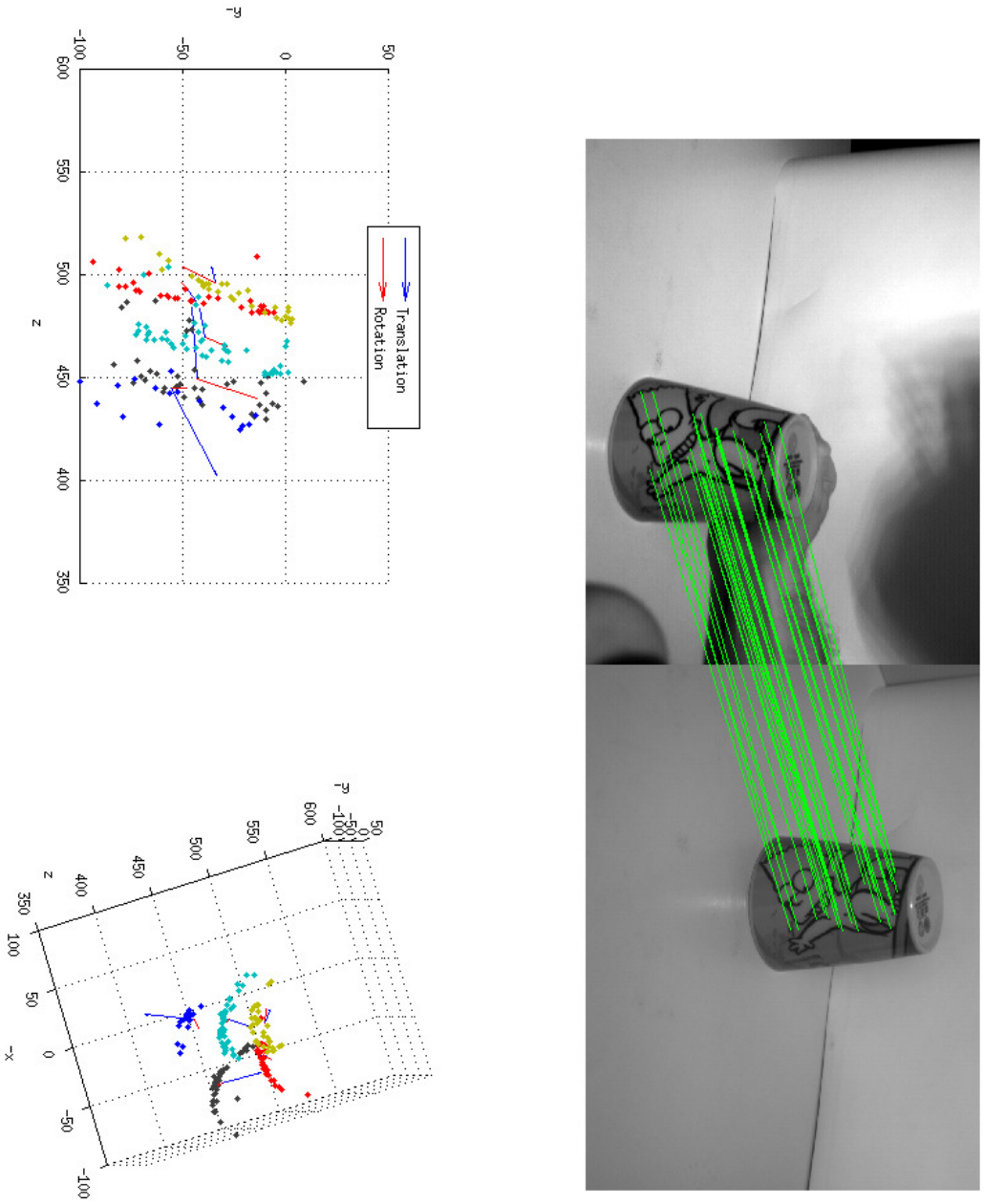


FIGURE VI.24 – La partie supérieure de l'image montre d'un côté l'image courante et de l'autre le gabarit. Les lignes vertes indiquent les correspondances qui ont été retenues par «RANSAC». La partie inférieure de l'image montre la position et le mouvement estimés de la tasse selon deux points de vue. Nous observons que les axes de rotation sont tous parallèles, ce qui est attendu : la rotation de la tasse se fait autour d'un axe normal au plan de la table. Les axes de translation sont aussi globalement tous parallèles, ce qui est également prévisible puisque la translation se fait sur le plan de la table.

CHAPITRE VII

CONCLUSION

VII.1 Généralités

Des applications médicales à la robotique en passant par la domotique et la sécurité, les caméras, ou autres systèmes d'imagerie, se trouvent de nos jours un peu partout. Ces systèmes ont pour but de représenter sous forme d'images informatiques une partie du monde réel. Parmi eux, les caméras «CMOS» utilisant une acquisition séquentielle sont de plus en plus fréquentes depuis une dizaine d'années notamment en raison de l'amélioration de leur rapport signal sur bruit. Nous pouvons en trouver dans beaucoup de systèmes embarqués utilisant une caméra comme les téléphones portables ou tablettes car la technologie «CMOS» qui les compose est plus compacte, plus économe en énergie et plus simple donc moins coûteuse. Ce type de périphériques mobiles grand public est de plus en plus souvent amené à réaliser des opérations impliquant de la vision par ordinateur : cartographie et reconstruction tridimensionnelle, positionnement, réalité virtuelle, ...

De telles applications peuvent être perturbées par le fonctionnement de ces capteurs. En effet, avec les caméras «CMOS» utilisant une acquisition séquentielle, lorsque ce qui est observé se déforme ou se déplace, ce type de capteur produit des déformations dans l'image résultante. Dans cette thèse, nous avons présenté le mode de fonctionnement de ces caméras et explicité la cause de ces déformations. Ce phénomène, appelé l'effet «Rolling Shutter», a longtemps été considéré comme un défaut. De nombreuses méthodes ont été proposées pour corriger ce type d'artefact, nous en avons brièvement présentées quelques unes. Au lieu de corriger ces déformations, nous avons abordé la thématique du calcul de pose dynamique à partir de telles caméras, c'est-à-dire l'estimation à partir d'une image de la position, l'orientation et le mouvement d'un objet dans la scène filmée. Il suffit pour cela de connaître le modèle tridimensionnel de l'objet et de supposer le mouvement uniforme. Ces travaux sont une extension de

travaux précédents (Meingast et al., 2005; Ait-Aider et al., 2006).

VII.2 Travaux effectués

VII.2.1 Modélisation non-uniforme

En généralisant ces travaux précédents, nous avons nommé «calcul de pose dynamique» cette extraction d'information et proposé un cadre de travail générique permettant de réaliser aussi bien le calcul de pose statique habituel que le calcul de pose dynamique, et ceux avec différentes hypothèses : aucun mouvement, mouvement uniforme et mouvement lisse. Nous avons en effet proposé une nouvelle modélisation du mouvement permettant de lever la contrainte d'uniformité. Cette modélisation est basée sur une paramétrisation directe de la pose sur chaque ligne de l'image. En considérant les dérivées centrales finies de ces paramètres au cours du temps, et donc des lignes, nous pouvons imposer des contraintes de lissage en cherchant à les rendre nulles. Ce cadre de travail générique réalisant une optimisation non-linéaire sous contrainte, nous avons du proposer une méthode d'initialisation qui permette d'être proche de la solution recherchée. Cette méthode repose sur une estimation «Global Shutter» par morceaux. L'image est ainsi partitionnée en plusieurs bandes horizontales contenant au moins six points et pour lesquels un calcul de pose sera réalisé avec «EPnP» (Lepetit et al., 2009). Cette méthode d'initialisation est en temps linéaire par rapport au nombre de points dans l'image.

VII.2.2 Modélisation polynomiale

Ces travaux et les travaux précédents reposent sur une optimisation non-linéaire locale qui nécessite une bonne initialisation. Une autre approche est possible via des méthodes d'optimisation globale qui ne nécessitent pas d'initialisation et qui permettent de trouver de manière certifiée le minimum global de la fonction de coût. Bien qu'il n'existe pas de méthode d'optimisation globale d'une fonction de coût non-linéaire générale, dans le cas d'une fonction de coût polynomiale, de telles méthodes existent (Kukelova et al., 2008; Henrion et al., 2009). Nous avons donc développé et présenté deux modèles polynomiaux du phénomène «Rolling Shutter». Ces modèles reposent sur une légère hypothèse simplificatrice des travaux précédents (Ait-Aider et al., 2006) et travaillent en deux temps : estimation des paramètres de rotation en premier à partir d'une fonction de coût polynomiale, puis estimation des paramètres de translation ensuite. L'estimation des paramètres de rotation peut être réalisée de manière efficace en utilisant les capacités de calculs parallèles des ordinateurs modernes. Cela nécessite néanmoins un choix rigoureux des outils utilisés et une adaptation des différentes boites à outils retenues pour travailler ensemble. L'assurance d'obtenir le minimum global sans initialisation nous a permis

de réaliser, dans le cas d'objets suffisamment texturés, une mise en correspondance automatique entre les points d'intérêt de l'image et le modèle tridimensionnel de l'objet. L'existence de potentielles correspondances erronées a nécessité de rendre robuste l'estimation des paramètres. Cela a été réalisé en implémentant une version de l'algorithme «RANdom SAmple Consensus» (Fischler and Bolles, 1981). Dans cette implémentation, nous utilisons sept points comme modèle minimal et le nombre d'itérations est ajusté automatiquement en fonction du taux de correspondances erronées estimé.

VII.2.3 Expérimentations

Nous avons présenté également deux méthodes permettant la création de données simulées nécessaire à l'évaluation des travaux présentés. Grâce aux données simulées ainsi obtenues, nous avons pu comparer nos travaux avec les travaux précédents sur des mouvements uniformes ou non-uniformes. Différents aspects ont été ainsi étudiés : comportement par rapport au nombre de points utilisés pour l'estimation, comportement face au bruit ou encore face à des correspondances erronées.

Dans le cas de mouvements simulés non-uniformes, il est naturellement attendu que le modèle non-uniforme permette d'estimer le mouvement avec une meilleure précision que les travaux précédents ou que le modèle polynomial. C'est ce qui est constaté lorsque le degré des contraintes de lissage est choisit de manière adéquate. Il faut cependant noter que cette méthode requiert un nombre conséquent de points mis en correspondances, de sorte que leur répartition le long des lignes soit suffisamment dense. Le modèle polynomial donne dans ce cadre des résultats similaires aux travaux précédents au niveau de la précision sur les paramètres de rotation. En revanche la précision sur les paramètres de translation est fortement impactée par l'opération de récupération de ces derniers à partir des paramètres de rotation.

Lors des expérimentations avec le jeu de données contenant des mouvements uniformes, il a été constaté que le modèle polynomial donne des résultats comparables aux travaux précédents en ce qui concerne les paramètres de translation. En revanche la précision sur l'axe de la rotation est nettement améliorée. Ceci s'explique par le fait que les paramètres de rotation sont ceux estimés lors l'optimisation globale. Le modèle non-uniforme peine pour sa part à donner des résultats satisfaisants. De plus, il est extrêmement sensible au bruit notamment lorsque celui-ci dépasse quelques pixels. La robustesse de la mise en correspondance automatique a également été testée sur ce jeu de données. Nous avons remarqué que la précision est globalement stable quelque soit le taux de correspondances erronées introduites, ces dernières étant bien filtrées par «RANSAC». Il est en revanche très difficile de régler les différents seuils de l'algorithme.

Deux expériences sur des données réelles dont les correspondances étaient connues ont permis de confirmer que le mouvement estimé avec le modèle polynomial était similaire aux travaux précédents et conforme au mouvement réel de l'objet : translation pure pour l'une des séquences, et rotation autour d'un axe pour l'autre. Avec des objets texturés, deux expériences sur des données réelles avec corres-

Modèle	Uniforme	Uniformité interpolée par morceaux	Non-uniforme lisse de degré 2	Polynomial
Acronyme	«URS»	«PWGS»	«SLWRS2»	«PURS»
Publication	ECCV (Ait-Aider et al., 2006)	3DPVT (Magerand and Bartoli, 2010)	3DPVT (Magerand and Bartoli, 2010)	ECCV (Magerand et al., 2012)
Mouvement	Uniforme	Uniforme par morceaux	Lisse de degré 2	Uniforme
Optimisation	Non-linéaire, itérative, locale	Résolution de systèmes linéaires	Non-linéaire, itérative, locale	Polynomiale, globale
Contraintes	Relaxation empirique	Non-concerné	Relaxation automatique	Respectées
Correspondances	Connues	Connues	Connues	Automatiques
Répartition nécessaire des projections	Éparse	Dense	Dense	Éparse
Temps	Jusqu'à 1 sec si mauvaise initialisation ou implémentation	$<<$ 1 sec (constant, quasi-linéaire)	de 5 à 10 sec	$\approx$ 5 sec (constant)

FIGURE VII.1 – Synthèse des différentes méthodes et travaux autour de la thématique du calcul de pose dynamique avec les caméras CMOS utilisant une acquisition séquentielle et présentées dans ce document.

pondances inconnues ont été réalisées. Dans les deux cas, le mouvement estimé est conforme à celui de l'objet. Nous avons cependant remarqué une fois de plus la difficulté à régler les seuils de l'algorithme : des faux-positifs ont perturbé l'estimation du mouvement dans l'une des deux séquences.

VII.2.4 Synthèse

Le tableau présenté figure VII.1 montre un récapitulatif de tous les travaux sur la thématique du calcul de pose dynamique avec les caméras CMOS utilisant une acquisition séquentielle. Toutes les méthodes, et leur modèle, qui ont été présentés jusqu'ici dans ce document.

VII.3 Perspectives

VII.3.1 À court terme

Il reste de nombreux problèmes à résoudre, tant dans les travaux que nous venons de présenter que de manière plus large. Quelques pistes relativement abordables peuvent déjà être proposées :

- Les faux-positifs lors de la mise en correspondance automatique viennent ils des erreurs dans le modèle 3D ou d'une ambiguïté plus profonde dans le modèle polynomial (objet coplanaire par exemple) ?
- Quelle est l'origine de l'erreur élevée sur les translations avec le modèle polynomial ? Est ce vraiment l'accumulation des erreurs due à la succession des deux estimations ?
- Poursuivre sur le modèle uniforme par morceaux avec interpolation pour arriver à du «Structure from Motion» comme le «Rolling Shutter Bundle Adjustement» (Hedborg et al., 2012) ?

VII.3.2 À moyen terme

Le recours à la résolution d'un problème de programmation semi-définie pour minimiser globalement, et sous contrainte, la fonction de coût du modèle uniforme polynomial est très coûteuse en terme de temps de calcul, et ce même avec l'utilisation des capacités de calculs parallèles des machines modernes. Une alternative à cette minimisation est la résolution du système polynomial obtenu en dérivant cette dernière par rapport aux différentes variables. L'utilisation de méthodes de résolution algébriques telles que celles présentées dans (Byröd et al., 2009; Kukelova et al., 2008) permettent de réaliser cette résolution de manière beaucoup plus rapide en obtenant tous les points stationnaires (où le gradient s'annule) avec quelques calculs algébriques et décompositions matricielles, quitte à devoir ensuite les filtrer pour trouver parmi eux le minimum global. Cela permettrait d'arriver à une méthode d'estimation temps réel. De plus, nous ne nous sommes pas intéressés à la détermination du cas minimal.

Il existe également de nombreux autres systèmes d'imagerie sur lesquels peut se trouver un capteur dont l'acquisition est séquentielle. Par exemple le système Kinect (première version) utilise un tel capteur, ce qui génère des déformations lors de la reconstruction tridimensionnelle si l'objet se déforme ou se déplace. Plutôt que de chercher à corriger ces déformations, il serait possible d'en tenir compte pour estimer des paramètres du mouvement afin de les utiliser dans les applications, d'autant que le système permet l'obtention du modèle tridimensionnel avec les correspondances. Nous pouvons aussi citer l'échographie tridimensionnelle cardiaque qui est obtenue par une acquisition séquentielle de plusieurs échographies bidimensionnelles observant chacune une coupe de la zone d'intérêt. Pour éviter les effets de déformation dus à l'acquisition asynchrone de ces coupes, elles sont réalisées en étant synchronisées sur les battements du coeur. Elles sont ensuite juxtaposées pour former l'échographie tridimensionnelle, qui représente alors le coeur comme un objet indéformable obtenu à un instant donné. Nous avons en effet supposé dans tous les travaux présentés que l'objet observé était rigide. Ici cette hypothèse est

faussée durant l'acquisition si la synchronisation avec les battements du coeur est supprimée, il serait alors intéressant d'étudier la problématique des objets immobiles mais déformables. Dans ce cas, l'information contenue dans l'image tridimensionnelle résultante reflète alors les battements du coeur et pourrait permettre d'extraire leurs paramètres (vitesse de contraction par exemple). Il serait possible d'obtenir une imagerie des mouvements du coeur qui soit plus rapide que les méthodes actuelles, une seule et uniquement acquisition asynchrone étant nécessaire.

VII.4 Publications

Les travaux présentés dans cette thèse ont été publiés dans plusieurs conférences, une nationale et deux internationales :

- Les prémices des travaux sur la modélisation non-uniforme ont été présentés lors de la conférence Orasis 2009 sous le titre «Un modèle de projection «Rolling Shutter» flexible appliqué au calcul de pose» ;
- Les travaux complets sur la modélisation non-uniforme ont été présentés lors de la conférence 3DPVT 2010 sous le titre «A Generic Rolling Shutter Camera Model and its Application to Dynamic Pose Estimation» ;
- Les travaux sur le modèle uniforme polynomial ont été présentés lors de la conférence ECCV 2012 sous le titre «Global Optimization of Object Pose and Motion from a Single Rolling Shutter Image with Automatic 2D-3D Matching».

ANNEXE A

OPTIMISATION NUMÉRIQUE

L'optimisation numérique est la branche des mathématiques s'intéressant au problème qui consiste à trouver la valeur d'un ensemble minimisant ou maximisant une fonction de cet ensemble vers l'ensemble des nombres réels. Cette valeur peut être déterminée analytiquement ou trouvée numériquement par approximation successive. Lorsque l'ensemble des valeurs admissibles comme solution est restreint, par exemple par des contraintes d'inégalité, l'optimisation est alors dite sous contrainte. Dans cette annexe quelques unes des méthodes d'optimisation les plus courantes en vision par ordinateur et utilisées dans cette thèse sont présentées.

A.1 Moindres carrés linéaires

La méthode des moindres carrés linéaires s'applique pour les problèmes qui s'expriment sous la forme d'un système linéaire des variables optimisées [1]. Ce type de problème a été très étudié et la littérature est abondante. Toutes les justifications et références nécessaires peuvent être obtenues dans (Axler, 1997; Hartley and Zisserman, 2003; Leon, 2006).

A.1.1 Définition du problème

Un systèmes d'équations linéaires peut s'écrire sous forme matricielle selon $\mathsf{M}\mathbf{x} = \mathbf{b}$ où $\mathbf{x}$ est le vecteur des inconnues, M est la matrice des coefficients des inconnues et $\mathbf{b}$ est le vecteur des termes

1. Il faut noter que bien que le problème doit être linéaire par rapport aux variables optimisées, il n'est pas nécessaire qu'il soit linéaire par rapport aux données. L'optimisation des coefficients d'un polynôme est par exemple un problème linéaire sur les variables à optimiser mais non-linéaire sur les données.

constants. La taille de M est de $m \times n$, où m est le nombre d'équations du système et n le nombre d'inconnues.

Le nombre d'équations et de variables optimisées conditionne l'existence de solutions, ainsi que le rang de la matrice M. Ce dernier dépend de la présence de dépendances linéaires entre les inconnues ou entre les équations. Lorsque la matrice est de rang plein, $\mathrm{rang}(\mathsf{M}) = \min(m, n)$, alors trois cas peuvent se présenter qui déterminent s'il existe des solutions et la méthode à appliquer pour les trouver :

- Il y a moins d'équations que d'inconnues ($m < n$), alors il existe plusieurs solutions qui forment un sous-espace vectoriel de $\mathbb{R}^n$ de dimension $n - m$.
- Il y a exactement autant d'équations que d'inconnues ($m = n$), alors il existe une unique solution tant que M est inversible.
- Il y a plus d'équations que d'inconnues ($m > n$), alors il n'existe de solution exacte que si $\mathbf{b}$ appartient à l'image de M. Généralement ce n'est pas le cas, néanmoins il existe toujours une solution qui minimise l'erreur entre $\mathsf{M}\mathbf{x}$ et $\mathbf{b}$. La fonction de coût $f(\mathbf{x})$ associée à ces problèmes est ainsi donnée par la norme au carré du vecteur des résidus $\mathbf{e}(\mathbf{x})$:

$$f(\mathbf{x}) = \| \underbrace{\mathsf{M}\mathbf{x} - \mathbf{b}}_{=\mathbf{e}(\mathbf{x})} \|^2. \tag{A.1}$$

Le vecteur $\mathbf{b}$ peut être le vecteur nul, ce qui forme un cas particulier. On parle alors du système homogène $\mathsf{M}\mathbf{x} = \mathbf{0}$. Lorsque qu'il y a autant ou plus d'équations que d'inconnues et que M est de rang plein, il n'y a qu'une solution triviale consistant à prendre $\mathbf{x} = \mathbf{0}$. Cependant si la matrice M est singulière et qu'il y a plus d'équations que d'inconnues, il peut exister une infinité de solutions non triviales (pour toute solution $\mathbf{x}$, le vecteur $\lambda\mathbf{x}$, $\lambda \in \mathbb{R}^*$ est aussi solution).

A.1.2 Méthodes de résolution

A.1.2.1 Autant d'équations que d'inconnues

L'approche la plus simple serait de calculer, lorsqu'elle existe, l'inverse M^{-1} de la matrice M et alors $\mathbf{x} = \mathsf{M}^{-1}\mathbf{b}$. Cette approche revient à résoudre directement le système d'équations en utilisant la méthode de l'élimination de Gauss-Jordan (ou pivot de Gauss). Cette méthode pose néanmoins deux problèmes :

- La complexité algorithmique de l'élimination de Gauss-Jordan dans le cas général est asymptotiquement en $O(n^3)$, ce qui provoque très vite des problèmes de performance.
- L'élimination de Gauss-Jordan procède à de nombreux calculs en nombres flottants dont une bonne part de divisions et multiplications. Ces calculs sont par nature de précision limitée, par conséquent la stabilité numérique de cette méthode est incertaine.

Une méthode alternative, plus performante et plus stable numériquement, consiste à utiliser la décomposition LU présentée dans l'annexe B.1 : $\mathsf{M} = \mathsf{LU}$. Nous avons alors $\mathsf{LU}\mathbf{x} = \mathbf{b}$, et en posant $\mathbf{y} = \mathsf{U}\mathbf{x}$, nous obtenons alors deux systèmes d'équations triangulaires qui sont résolus simplement par substitution inverse : $\mathsf{L}\mathbf{y} = \mathbf{b}$ qui est résolu pour trouver $\mathbf{y}$ et $\mathsf{U}\mathbf{x} = \mathbf{y}$ qui est résolu pour trouver $\mathbf{x} = \mathsf{U}^\top\mathbf{y}$ puisque U est orthogonale.

A.1.2.2 Plus d'équations que d'inconnues

Dans ce cas nous cherchons le minimum de la fonction de coût $f(\mathbf{x})$ présentée à l'équation (A.1). Cette fonction de coût est convexe et ne possède qu'un unique minimum. Celui-ci se trouve à l'endroit où les dérivées partielles de $f(\mathbf{x})$ s'annulent. Après dérivation, nous obtenons alors le système

$$\mathsf{M}^\top(\mathsf{M}\mathbf{x} - \mathbf{b}) = 0 \Leftrightarrow \mathsf{M}^\top\mathsf{M}\mathbf{x} = \mathsf{M}^\top\mathbf{b} \tag{A.2}$$

connu comme étant les équations normales. Ce système fait intervenir autant d'équations que d'inconnues, il peut donc être résolu comme précédemment en inversant $\mathsf{M}^\top\mathsf{M}$, ce qui donne $\mathbf{x} = (\mathsf{M}^\top\mathsf{M})^{-1}\mathsf{M}^\top\mathbf{b}$. La matrice $(\mathsf{M}^\top\mathsf{M})^{-1}\mathsf{M}^\top$ est une pseudo-inverse de la matrice M, généralement notée M^+. Il est bien évident que pour les mêmes raisons de performance et de stabilité que précédemment, nous ne calculerons jamais réellement cette matrice à cause de l'inversion. Les équations normales seront résolues avec une décomposition LU ou de Cholesky de la matrice $\mathsf{M}^\top\mathsf{M}$.

Une autre méthode plus stable numériquement existe en utilisant la décomposition en valeurs singulières (SVD) présentée en annexe B.3 : $\mathsf{M} = \mathsf{USV}^\top$. La fonction de coût de l'équation (A.1) s'écrit alors

$$f(\mathbf{x}) = \|\mathsf{USV}^\top\mathbf{x} - \mathbf{b}\|^2 = \|\mathsf{SV}^\top\mathbf{x} - \mathsf{U}^\top\mathbf{b}\|^2 \tag{A.3}$$

puisque la matrice U est orthogonale. En posant $\mathbf{y} = \mathsf{V}^\top\mathbf{x}$ et $\mathbf{d} = \mathsf{U}^\top\mathbf{b}$, nous obtenons alors $f(\mathbf{x}) = \|\mathsf{S}\mathbf{y} - \mathbf{d}\|^2$. Comme la matrice S est diagonale, son inverse est facile à calculer tant que les valeurs singulières de M sont toutes non nulles, ce qui est le cas si elle est de rang plein. Il est alors évident que $\mathbf{y} = \mathsf{S}^{-1}\mathbf{d}$ d'où il découle que $\mathbf{x} = \mathsf{VS}^{-1}\mathbf{d}$ puisque V est orthogonale. Finalement nous obtenons la solution $\mathbf{x} = \mathsf{VS}^{-1}\mathsf{U}^\top\mathbf{b}$. La matrice $\mathsf{VS}^{-1}\mathsf{U}^\top$ est aussi une pseudo-inverse de la matrice M. Lorsque la matrice M est singulière (ou très proche de l'être), la matrice diagonale S contient des entrées diagonales nulles (ou proche de l'être). Une solution, potentiellement plus stable numériquement, peut être obtenue dans ce cas en mettant à 0 les entrées correspondantes de la matrice S^{-1}.

A.1.2.3 Système homogène

Dans le cas d'un système homogène $\mathsf{M}\mathbf{x} = \mathbf{0}$, nous cherchons habituellement une solution autre que la solution triviale $\mathbf{x} = \mathbf{0}$. Il faut alors imposer une contrainte sur $\mathbf{x}$ comme $\|\mathbf{x}\| = 1$, qui permettra de trouver une solution minimisant $\|\mathsf{M}\mathbf{x}\|$. Lorsque la matrice M est singulière, les propriétés de la décomposition en valeurs singulières permettent, en décomposant $\mathsf{M} = \mathsf{U}\mathsf{S}\mathsf{V}^\top$, de trouver une base de son noyau qui est constituée des vecteurs singuliers à droite contenus dans V et correspondant aux valeurs singulières nulles. Si la matrice est de rang plein, il est possible de montrer que le noyau est vide mais que les vecteurs singuliers à droite correspondants aux plus petites valeurs singulières sont ceux qui minimisent $\|\mathsf{M}\mathbf{x}\|$. Ainsi si l'hypothèse est faite que la décomposition a été réalisée en triant les valeurs singulières de la plus grande à la plus petite, alors la dernière colonne de V contient un vecteur normé qui minimise $\|\mathsf{M}\mathbf{x}\|$.

A.2 Méthodes polynomiales globales

Lorsque la fonction de coût optimisée n'est pas linéaire mais reste néanmoins polynomiale, il existe différentes méthodes permettant de trouver les paramètres qui aboutissent au minimum global. Les deux méthodes présentées ici permettent l'optimisation sous contraintes, elles aussi polynomiales. Ces deux méthodes aboutissent à la résolution d'un problème impliquant une optimisation dans l'ensemble des matrices semi-définies positives. Plus de détails sur ces différentes méthodes et notions se trouvent dans (Laurent, 2009). Un problème d'optimisation polynomiale sous contraintes écrit sous sa forme la plus générale est donné par

$$\min_{\mathbf{x}\in\mathbb{R}^\mathbf{n}} p(\mathbf{x}) \quad s.c. \quad q_1(\mathbf{x}) \geq 0, \ldots, q_m(\mathbf{x}) \geq 0, \tag{A.4}$$

où $p(\mathbf{x}) \in \mathbb{R}[\mathbf{x}]$ est la fonction de coût polynomiale et $q_1(\mathbf{x}), \ldots, q_m(\mathbf{x}) \in \mathbb{R}[\mathbf{x}]$ les contraintes polynomiales.

A.2.1 Programmation semi-définie

Des problèmes de programmation semi-définie (ou «Semi-Definite Programming», SDP) ont des applications dans de nombreux domaines des mathématiques appliquées : recherche opérationnelle, analyse de graphes, optimisation combinatoire, *etc.* En particulier, il est possible de relaxer certaines catégories de problèmes d'optimisation polynomiale à la résolution d'une suite de problèmes SDP. Les sections A.2.2 et A.2.3 décrivent deux méthodes de relaxation.

Une matrice carrée A de taille n est dite semi-définie positive si pour tout vecteur $\mathbf{x} \in \mathbb{R}^n$, nous

avons $\mathbf{x}^\top \mathsf{A}\mathbf{x} \geq 0$. Il est possible de montrer que cela est équivalent à l'existence d'une matrice B telle que $\mathsf{A} = \mathsf{B}^\top\mathsf{B}$. Autrement dit, il existe des vecteurs (les colonnes de B : $\mathbf{b}_i, i \in [1, n]$) tels que $a_{i,j} = \mathbf{b}_i \cdot \mathbf{b}_j$. Dans le cas où A est définie positive (c'est-à-dire $\mathbf{x}^\top \mathsf{A}\mathbf{x} > 0, \forall \mathbf{x} \in \mathbb{R}^n$), alors la matrice B peut être obtenue par la décomposition de Cholesky de A (voir annexe B.1). Notons $\mathcal{S}_n$ l'ensemble des matrices semi-définies positives et définissons le produit scalaire de deux éléments A et B de $\mathcal{S}_n$ par $\mathsf{A} \cdot \mathsf{B} = \mathrm{tr}(\mathsf{A}^\top\mathsf{B})$.

Un problème de programmation semi-définie est un problème d'optimisation sous contraintes de la forme

$$\min_{\mathsf{X}\in\mathcal{S}_n} \mathsf{C} \cdot \mathsf{X} \quad s.c. \quad \mathsf{A}_k \cdot \mathsf{X} \leq b_k, k \in [1, m], \tag{A.5}$$

où $\mathsf{C} \in \mathcal{S}_n$ est une matrice de coefficients décrivant les données du problème, la matrice X est une matrice semi-définie positive contenant les inconnues du problèmes, les m matrices $\mathsf{A}_k \in \mathcal{S}_n$ et coefficients b_k sont des données du problème et décrivent ses contraintes. Il est possible de se ramener à des contraintes d'égalité en augmentant la matrice X de m variables différences sur la diagonale (les autres nouvelles entrées sont toutes nulles).

Ce problème est non-linéaire et non lisse (les dérivées ne sont pas nécessairement calculables) mais reste néanmoins convexe. En effet, la fonction optimisée est linéaire et l'ensemble des solutions admissibles par les contraintes est convexe puisque les bordures de cet ensemble sont définies par des morceaux de courbes polynomiales. La non différentiabilité apparaît aux points de jointure de ces morceaux.

Il existe une dualité entre ce problème et le problème suivant :

$$\max_{\mathbf{y}\in\mathbb{R}^m} \mathbf{b} \cdot \mathbf{y} \quad s.c. \quad \sum_{k=1}^{m} (y_k \mathsf{A}_k) - \mathsf{C} \in \mathcal{S}_n, \tag{A.6}$$

où $\mathbf{b}$ est le vecteur contenant les coefficients des contraintes du problème original. Cette dualité est en règle générale faible, c'est-à-dire que la valeur maximale obtenue en résolvant le problème dual est un minimum de la valeur obtenue en résolvant le problème original. Sous certaines conditions particulières (dites de Slater) la dualité devient forte, c'est-à-dire que les deux valeurs obtenues en résolvant les deux problèmes sont égales.

Des méthodes de résolution existent tant pour le problème original que pour son dual. Néanmoins les méthodes les plus efficaces sont celles qui alternent la résolution du problème original et de son dual. C'est ce qui est réalisé dans (Vandenberghe and Boyd, 1996) qui est à la base de tous les outils modernes de programmation semi-définie. Ces méthodes de résolution produisent non seulement les paramètres estimés comme étant la solution optimale, mais aussi un certificat prouvant que c'est effectivement une solution optimale. Des boîtes à outils (Borchers, 1999; Sturm, 2001) existent pour venir à bout de ces problèmes. Une version adaptée au calcul parallèle de (Borchers, 1999) existe (Borchers and Young,

2007). C'est actuellement un des outils les plus efficaces et c'est cette librairie qui a été utilisée dans les travaux présentés ici, après quelques optimisations lors de la compilation.

A.2.2 La méthode «Sums Of Squares»

Un polynôme $h(\mathbf{x}) \in \mathbb{R}[\mathbf{x}]$ de degré $2d, d \in \mathbb{N}$ est dit «Sums Of Squares» (SOS) s'il satisfait les deux conditions suivantes :

- C'est un polynôme homogène, c'est-à-dire que tous ses monômes sont de degré $2d$;
- Il existe des polynômes homogènes $g_i(\mathbf{x}), i \in [1, k], k \geq 1$ de degré d tels que $h(x) = \sum_{i=1}^{k} g_i(\mathbf{x})^2$.

De manière équivalente, il est possible de montrer que tout polynôme homogène de degré $2d$ qui peut s'écrire sous la forme $h(\mathbf{x}) = \mathbf{z}^\top \mathsf{Q}\mathbf{z}$ où $\mathbf{z}$ est un vecteur de monômes de degré d, et où Q est une matrice symétrique [2] et semi-définie positive de coefficients, est un polynôme SOS. Lorsqu'un polynôme n'est pas homogène, on dit qu'il est SOS lorsque son homogénéisation $\tilde{p} \in \mathbb{R}[\mathbf{x}, x_{n+1}]$ [3] est SOS. Notons dorénavant $\mathcal{SOS}$ l'ensemble des polynômes SOS.

Résoudre un problème d'optimisation polynomiale tel que défini équation (A.4) revient tout naturellement à trouver

$$\max_{t,\mathbf{x}} t \quad \textit{s.c.} \quad p(\mathbf{x}) - t \geq 0, q_1(\mathbf{x}) \geq 0, \ldots, q_m(\mathbf{x}) \geq 0 \tag{A.7}$$

ce qui reste un problème difficile à résoudre. L'idée de la méthode SOS est alors de remplacer certaines conditions de positivité par des conditions d'appartenance à $\mathcal{SOS}$, tout polynôme SOS étant nécessairement positif quelque soit $\mathbf{x} \in \mathbb{R}^n$. La relaxation de ce problème en un problème d'optimisation SOS sous contraintes s'écrit alors sous la forme

$$\max_{t,\mathbf{x}} t \quad \textit{s.c.} \quad p(\mathbf{x}) - t = s_0(\mathbf{x}) + \sum_{j=1}^{m} s_j(\mathbf{x}) g_j(\mathbf{x}), \tag{A.8}$$

où $s_i(\mathbf{x}), i \in [0, m]$ sont SOS.

La valeur obtenue en résolvant ce problème est un minimum de la valeur obtenue en résolvant le problème sans relaxation (c'est-à-dire avec les contraintes de positivités). Sous certaines conditions sur les contraintes, l'égalité des deux valeurs peut arriver. Il est à noter que dans la formulation précédente, le degré des polynômes $s_i(\mathbf{x}), i \in [0, m]$ n'est pas restreint. Plus le degré est élevé et meilleure est l'estimation de t et donc la solution $\mathbf{x}$, mais aussi plus grand devient le problème SDP sous-jacent. Il est donc déraisonnable de choisir un degré trop élevé. Le degré minimal théorique est d tel que $2d$ soit

2. La symétrie de cette matrice implique son unicité.

3. L'homogénéisation $\tilde{h}(\mathbf{x}, x_{n+1})$ est obtenue en ajoutant une indéterminée par laquelle est multiplié chaque monôme autant de fois que nécessaire pour que tous les monômes soit de degré $2d$, et en ajoutant une contrainte $x_{n+1} = 1$.

supérieur ou égal au degré le plus élevé des polynômes $p, g_1, \ldots, g_m$. Mais s'il s'avère qu'en pratique ce degré n'est pas suffisant, il peut être augmenté jusqu'à obtention d'une solution satisfaisante.

Il existe une généralisation de ce problème SOS qui s'écrit sous la forme

$$\min_{\mathbf{u}\in\mathbb{R}^n} \mathbf{w}^\top \mathbf{u} \quad s.c. \quad a_{k,0}(\mathbf{x}) + \sum_{j=1}^{n} a_{k,j}(\mathbf{x}) u_j \in \mathcal{SOS}, \forall k \in [1, m], \tag{A.9}$$

où $\mathbf{u}$ est le vecteur des indéterminées. Les données du problème sont le vecteur de coefficients $\mathbf{w}$ et les polynômes $a_{k,j}, k \in [1, m], j \in [0, n]$. Pour que ce problème possède une solution, il faut qu'il existe une matrice semi-définie positive qui permet d'exprimer les polynômes $a_{k,0}(\mathbf{x}) + \sum_{j=1}^{n} a_{k,j}(\mathbf{x}) u_j$ comme des SOS. Pour plus de détails sur cette généralisation et l'utilisation des SOS pour résoudre les problèmes d'optimisation polynomiale, il est possible de se référer à (Parrilo, 2003) qui décrit la méthode plus en profondeur. Il existe une boîte à outils (Prajna et al., 2004) qui correspond à cette méthode. Pour des raisons de performance, cette boite à outils a été modifiée pour qu'elle soit compatible avec la librairie SDP mentionnée section A.2.1.

A.2.3 La méthode des moments

En analyse mathématique le moment d'ordre $r \in \mathbb{N}$ d'une mesure $f : \mathcal{I} \to \mathbb{R}$ (où $\mathcal{I} \subset \mathbb{R}$ est non réduit à un point et est appelé support de la mesure) est défini par la quantité

$$m_r = \int_{\mathcal{I}} x^r f(x) \mathrm{d}x. \tag{A.10}$$

Cette quantité n'est définie que si $x^r f(x)$ est intégrable. Dans le cas contraire, le moment d'ordre r de f n'existe pas. Il faut bien différencier l'existence d'un moment et le fait qu'un moment soit nul.

Étant donné une suite de moments (m_r), il est possible de se poser la question de l'existence d'une mesure dont les moments correspondent à la suite donnée, il s'agit du problème des moments. La réponse à cette question est donnée par l'analyse des matrices de Hankel (ou matrice des moments) associées à la suite des moments :

$$\mathsf{H}_n = \begin{bmatrix} m_0 & m_1 & \cdots & m_{n-1} & m_n \\ m_1 & m_2 & \cdots & m_n & m_{n+1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ m_{n-1} & m_n & \cdots & \ddots & \vdots \\ m_n & m_{n+1} & \cdots & \cdots & m_{2n} \end{bmatrix}, \tag{A.11}$$

où $h_{i,j} = m_{i+j}$. Si toutes ces matrices sont semi-définies positives alors il existe une mesure dont les

moments forment la suite (m_r). Cela permet de construire une hiérarchie de problèmes de programmation semi-définie dont la résolution démontre l'existence ou non de la mesure.

Dans le cas où il y a plusieurs mesures à estimer, ce problème s'écrit alors sous sa forme la plus générique

$$\min_{\mu,\mathbf{x}} \sum_k \int_{\mathcal{K}_k} g_{0,k}(\mathbf{x})\mu_k(\mathbf{x})\mathrm{d}\mathbf{x} \quad s.c. \quad \sum_k \int_{\mathcal{K}_k} h_{j,k}(\mathbf{x})\mu_k(\mathbf{x})\mathrm{d}\mathbf{x} \geq b_j, j \in [0,\dots] \tag{A.12}$$

où μ est le vecteur des mesures. Les ensembles $\mathcal{K}_k$ définissent chacun le support de la mesure μ_k :

$$\mathcal{K}_k = \{\mathbf{x} \in \mathbb{R}^n, g_{i,k}(\mathbf{x}) \geq 0, i \in [1,\dots]\}. \tag{A.13}$$

Les polynômes $g_{i,k}$ et $h_{j,k}$ ainsi que le vecteur des coefficients $\mathbf{b}$ sont les données du problème. Ce problème très large englobe le problème de l'optimisation polynomiale sous contraintes. En effet, en reprenant les notations de (A.4), il suffit de poser le problème des moments généralisé pour une seule mesure ($k = 1$) telle que

$$\begin{cases} g_{0,1}(\mathbf{x}) = p(\mathbf{x}) \\ g_{i,1}(\mathbf{x}) = q_i(\mathbf{x}), i \in [1,m] \\ h_{0,1}(\mathbf{x}) = 1, b_0 = 1 \\ h_{1,1}(\mathbf{x}) = 1, b_1 = -1 \end{cases} \tag{A.14}$$

pour obtenir l'existence ou non d'une ou plusieurs solutions.

Les articles (Lasserre, 2001, 2008) contiennent plus de détails sur la méthode des moments dans le cas général de l'équation (A.12). Dans le cas de l'optimisation polynomiale sous contraintes, il s'agit en fait de la méthode duale à la méthode SOS. (Lasserre, 2009; Laurent, 2009) donnent plus d'explication sur les raisons de cette dualité. Une implémentation de la méthode des moments est proposée par (Henrion et al., 2009). Après quelques modifications et adaptations présentées chapitre V, cette dernière peut utiliser nativement la librairie SDP mentionnée section A.2.1.

A.3 Méthodes non-linéaires locales

Cette section aborde l'optimisation d'une fonction de coût qui soit non-linéaire, avec ou sans contraintes sur les paramètres. Les méthodes présentées sont des méthodes itératives, c'est-à-dire qu'elles procèdent par raffinements successifs de la solution en partant d'une solution initiale. De plus, elles ne réalisent qu'une optimisation locale et ne peuvent donc converger que vers une solution se trouvant près de la solution initiale. Elles requièrent donc un soin particulier quant au choix de la solution initiale dans le cas d'un problème présentant de multiples minima locaux autres que le minimum global. Pour plus

de détails sur les deux méthodes abordées ici, il est possible de se référer à (Nocedal and Wright, 2006; Fletcher, 1987).

A.3.1 La méthode de Levenberg-Marquardt

Bien qu'étant une méthode assez ancienne, la méthode de Levenberg-Marquardt reste une des méthodes les plus utilisées pour l'optimisation des moindres carrés non-linéaires. Le principe de base de cette méthode remonte à (Levenberg, 1944), et a été ensuite perfectionné par (Marquardt, 1963; Moré, 1977). Elle permet la résolution des problèmes sur-déterminés de minimisation au sens des moindres carrés, comportant des fonctions non-linéaires, et qui s'expriment donc sous la forme

$$\min_{\mathbf{x}} \|\mathbf{f}(\mathbf{x}) - \mathbf{b}\|^2. \tag{A.15}$$

Cette méthode généralise donc en quelque sorte la méthode des moindres carrés linéaires. La méthode de Levenberg-Marquardt est le résultat de la combinaison de deux autres méthodes : Gauss-Newton et descente du gradient.

La méthode de Gauss-Newton utilise le développement limité à l'ordre premier de la fonction non-linéaire $\mathbf{f}(\mathbf{x})$ donné par

$$\mathbf{f}(\mathbf{x} + \delta\mathbf{x}) = \mathbf{f}(\mathbf{x}) + \mathsf{J}(\mathbf{x})\delta\mathbf{x}, \tag{A.16}$$

qui permet de linéariser localement la fonction de coût. La matrice $\mathsf{J}(\mathbf{x})$ est la matrice jacobienne de $\mathbf{f}(\mathbf{x})$ prise en $\mathbf{x}$, telle que $J_{i,j}$ soit la dérivée en $\mathbf{x}$ de $f_i(\mathbf{x})$ par rapport à x_j. L'idée de la méthode de Gauss-Newton est alors de calculer à chaque itération l'incrément $\delta\mathbf{x}$ qui minimise $\mathbf{f}(\mathbf{x} + \delta\mathbf{x}) - \mathbf{b}$. D'après (A.16), la valeur $\delta\mathbf{x}$ recherchée est alors celle minimisant $\mathbf{f}(\mathbf{x}) + \mathsf{J}(\mathbf{x})\delta\mathbf{x} - \mathbf{b}$ qui est un problème de minimisation linéaire par rapport à $\delta\mathbf{x}$. Ainsi nous obtenons l'incrément en résolvant les équations normales

$$\mathsf{J}(\mathbf{x})^\top \mathsf{J}(\mathbf{x})\delta\mathbf{x} = -\mathsf{J}(\mathbf{x})^\top(\mathbf{f}(\mathbf{x}) - \mathbf{b}) \Rightarrow \delta\mathbf{x} = -\mu\mathsf{J}(\mathbf{x})^+(\mathbf{f}(\mathbf{x}) - \mathbf{b}), \tag{A.17}$$

où μ est un coefficient permettant de fixer la longueur du pas. Cette méthode converge rapidement pour peu que la solution initiale soit suffisamment proche de la solution finale. Dans le cas contraire, elle peut ne pas converger.

Le principe de la méthode de descente du gradient est de mettre à jour $\mathbf{x}$ à chaque itération en se dirigeant vers la plus forte pente de la fonction de coût. Cela garantit d'aboutir à un minimum, même si cette méthode peut être excessivement lente. Afin de calculer la direction de la plus forte pente, l'approximation au premier ordre de $\|\mathbf{f}(\mathbf{x}) - \mathbf{b}\|^2$ est utilisée. Cette approximation peut être formée à

partir de la Jacobienne $\mathsf{J}(\mathbf{x})$. La direction est alors donnée par

$$\delta\mathbf{x} = -\xi\mathsf{J}(\mathbf{x})^\top(\mathbf{f}(\mathbf{x}) - \mathbf{b}), \tag{A.18}$$

où ξ est un coefficient qui permet d'ajuster la longueur de l'avancée le long de cette direction.

En comparant les deux incréments de (A.17) et de (A.18), il est possible de remarquer que l'incrément dans le cas de la méthode de descente du gradient approxime l'incrément de la méthode de Gauss-Newton. L'idée de l'algorithme de Levenberg-Marquardt est d'alterner entre ces deux incréments pour conserver le meilleur de chaque méthode : une descente assurée mais lente ou une convergence rapide mais incertaine. Afin de coupler les deux méthodes, les deux coefficients μ et ξ seront remplacés par un seul λ qui fixera la longueur du pas. L'incrément utilisé dans la méthode de Levenberg-Marquardt est alors donné par

$$\delta\mathbf{x} = -\Big(\mathsf{J}(\mathbf{x})^\top\mathsf{J}(\mathbf{x}) + \lambda\mathtt{I}\Big)^{-1}\mathsf{J}(\mathbf{x})^\top(\mathbf{f}(\mathbf{x}) - \mathbf{b}). \tag{A.19}$$

La valeur de λ joue un rôle capital : lorsqu'elle est faible l'incrément de Levenberg-Marquardt tend à être l'incrément de Gauss-Newton, assurant ainsi une convergence rapide bien qu'incertaine. Si l'incrément ainsi calculé ne permet pas de faire baisser la fonction de coût, il suffit d'augmenter λ pour se rapprocher d'avantage de la descente de gradient et donc d'assurer la descente. λ est ainsi augmenté sans changer la solution courante jusqu'à trouver un incrément qui permet une baisse de la fonction de coût. Lorsque c'est le cas, la solution courante est changée et on décroît à nouveau λ tant que la fonction de coût décroît, retrouvant ainsi une convergence rapide. Une bonne valeur d'initialisation de λ est donnée par la moyenne des éléments diagonaux de $\mathsf{J}(\mathbf{x})^\top\mathsf{J}(\mathbf{x})$. Habituellement le facteur de croissance et décroissance de λ est choisit empiriquement à 10. Nous encadrons également la valeur de λ pour qu'elle reste dans un certain intervalle, généralement $[10^{-9}; 10^{9}]$ de sorte à ne pas dépasser les capacités de calculs de la machine.

L'arrêt de l'optimisation est conditionné par plusieurs facteurs. Tout d'abord un nombre maximum d'itérations est généralement fixé afin d'avoir la certitude que l'algorithme s'arrêtera. L'arrêt est aussi provoqué lorsque $\|\mathbf{f}(\mathbf{x}) - \mathbf{b}\|^2$ est devenue inférieure à une certaine valeur dépendante du problème considéré. Enfin une dernière condition est donnée par la norme du pas $\|\delta\mathbf{x}\|$ qui, si elle est devenue trop petite, signifie que l'on ne fera plus de progrès car une zone de stabilité de la fonction de coût a été atteinte. L'algorithme complet est récapitulé dans la figure A.1.

A.3.2 La méthode «Sequential Quadratic Programming»

La programmation quadratique séquentielle (ou «Sequential Quadratic Programming») (Powell, 1983; Schittkowski, 1985; Fan et al., 1988) permet de résoudre des problèmes d'optimisation d'une fonction de coût non-linéaire sous des contraintes elles aussi non-linéaires. Ce type de problème s'écrit donc

$iter \leftarrow 0$;
Calculer l'erreur initiale $besterr \leftarrow \|\mathbf{f}(\mathbf{x}) - \mathbf{b}\|^2$;
Calculer la valeur initiale de λ comme la moyenne des éléments diagonaux de $\mathtt{J}(\mathbf{x})^\top \mathtt{J}(\mathbf{x})$;
tant que $iter \leq maxiter$ et $besterr \leq minerr$ et $\|\delta\mathbf{x}\| \geq minstep$ **faire**
  Calculer l'incrément $\delta\mathbf{x} = -\left(\mathtt{J}(\mathbf{x})^\top \mathtt{J}(\mathbf{x}) + \lambda \mathtt{I}\right)^{-1} \mathtt{J}(\mathbf{x})^\top (\mathbf{f}(\mathbf{x}) - \mathbf{b})$;
  Calculer l'erreur $err \leftarrow \|\mathbf{f}(\mathbf{x} + \delta\mathbf{x}) - \mathbf{b}\|^2$;
  si $err < besterr$ **alors**
    $\mathbf{x} \leftarrow \mathbf{x} + \delta\mathbf{x}$;
    $besterr \leftarrow err$;
    $\lambda \leftarrow \max(\frac{\lambda}{10}, 10^{-9})$;
  sinon
    $\lambda \leftarrow \min(10\lambda, 10^9)$;
  $iter \leftarrow iter + 1$;

FIGURE A.1 – Algorithme de Levenberg-Marquardt pour l'optimisation $\min_{\mathbf{x}} \|\mathbf{f}(\mathbf{x}) - \mathbf{b}\|^2$ d'une fonction de coût $\mathbf{f}(\mathbf{x})$ non-linéaire où $\mathbf{x}$ est le vecteur des paramètres. Les valeurs $minerr$ et $minstep$ sont respectivement les bornes d'arrêt de $\|\mathbf{f}(\mathbf{x}) - \mathbf{b}\|^2$ et $\|\delta\mathbf{x}\|$.

sous la forme

$$\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x}) \quad s.c. \quad \begin{cases} h_i(\mathbf{x}) = 0, i \in [1, m] \\ g_j(\mathbf{x}) \leq 0, j \in [1, p] \end{cases}, \tag{A.20}$$

où f est la fonction à minimiser, $\mathbf{h}$ et $\mathbf{g}$ sont respectivement les contraintes d'égalité et d'inégalité. Il est possible de ramener les contraintes d'inégalité à des contraintes d'égalité en introduisant des variables de différence, raison pour laquelle nous traitons par la suite les contraintes d'inégalité comme des contraintes d'égalité.

Pour résoudre ce problème, nous introduisons des multiplicateurs de Lagrange $\boldsymbol{\lambda} \in \mathbb{R}^m$ et $\boldsymbol{\mu} \in \mathbb{R}^p$ pour les contraintes. Nous nous intéressons alors au Lagrangien du problème défini par

$$L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = f(\mathbf{x}) - \boldsymbol{\lambda}^\top \mathbf{h}(\mathbf{x}) - \boldsymbol{\mu}^\top \mathbf{g}(\mathbf{x}). \tag{A.21}$$

Un point solution du problème doit vérifier les conditions nécessaires d'optimalité au premier ordre aussi appelées conditions de Karush–Kuhn–Tucker :

$$\nabla_{L(\mathbf{x},\boldsymbol{\lambda},\boldsymbol{\mu})} = 0 \Rightarrow \begin{bmatrix} \nabla_f(\mathbf{x}) - \mathtt{J}_{\mathbf{h}}(\mathbf{x})^\top \boldsymbol{\lambda} - \mathtt{J}_{\mathbf{g}}(\mathbf{x})^\top \boldsymbol{\mu} \\ \mathbf{h}(\mathbf{x}) \\ \mathbf{g}(\mathbf{x}) \end{bmatrix} = \mathbf{0}, \tag{A.22}$$

où $\nabla_f(\mathbf{x})$ désigne le gradient de f par rapport à $\mathbf{x}$ pris en $\mathbf{x}$ et $\mathtt{J}_{\mathbf{g}}(\mathbf{x})$ (respectivement $\mathtt{J}_{\mathbf{h}}(\mathbf{x})$) désigne

la matrice Jacobienne de $\mathbf{g}$ (respectivement $\mathbf{h}$) par rapport à $\mathbf{x}$ prise en $\mathbf{x}$.

Ce système de $n+m+p$ équations dont les inconnues sont $(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu})$ peut être résolu par la méthode de Newton-Raphson. Nous obtenons alors un incrément $\mathbf{d}$ de $(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu})$ qui doit satisfaire le système linéaire

$$\underbrace{\begin{bmatrix} \mathsf{H}_L(\mathbf{x}) & -\mathsf{G}(\mathbf{x}) \\ \mathsf{G}(\mathbf{x})^\top & 0 \end{bmatrix}}_{=\mathsf{A}} \mathbf{d} = - \begin{bmatrix} \nabla_{f(\mathbf{x})} - \mathsf{J}_{\mathbf{h}}(\mathbf{x})^\top \boldsymbol{\lambda} - \mathsf{J}_{\mathbf{g}}(\mathbf{x})^\top \boldsymbol{\mu} \\ \mathbf{h}(\mathbf{x}) \\ \mathbf{g}(\mathbf{x}) \end{bmatrix}, \tag{A.23}$$

où $\mathsf{H}_L(\mathbf{x})$ est la matrice Hessienne du Lagrangien par rapport à $\mathbf{x}$ prise en $\mathbf{x}$ et $\mathsf{G}(\mathbf{x})$ est la concaténation horizontale de $\mathsf{J}_{\mathbf{h}}(\mathbf{x})$ et $\mathsf{J}_{\mathbf{g}}(\mathbf{x})$. Si la matrice A n'est pas singulière, alors il est possible de résoudre ce système par élimination de Gauss-Jordan, ou tout autre méthode apparentée basée sur une décomposition matricielle. Cela reste néanmoins un problème de taille conséquente, en particulier lorsqu'il y a de nombreuses contraintes.

De plus il faut recalculer à chaque itération les matrices Jacobiennes et Hessiennes, ce qui peut être coûteux en temps de calcul, notamment si on ne dispose pas d'une expression analytique de ces dernières. Des méthodes comme BFGS (Broyden–Fletcher–Goldfarb–Shanno) (Broyden, 1970; Fletcher, 1970; Goldfarb, 1970; Shanno, 1970) ont été créées pour contourner cet inconvénient en utilisant une mise à jour de ces matrices plutôt que de les recalculer intégralement.

Il est possible, et généralement plus efficace, de calculer l'incrément $\mathbf{d}$ directement en réalisant une optimisation quadratique plutôt que de former et résoudre le système (A.23). En effet ce système a la même forme que celui obtenu en linéarisant les contraintes de (A.20) et en utilisant l'approximation quadratique du Lagrangien :

$$\min_{\mathbf{d}} L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) + \nabla_L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu})^\top \mathbf{d} + \frac{1}{2} \mathbf{d}^\top \nabla_L^2(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \mathbf{d} \quad s.c. \quad \begin{cases} \mathsf{J}_{\mathbf{h}}(\mathbf{x})^\top \mathbf{d} + \mathbf{h}(\mathbf{x}) = \mathbf{0} \\ \mathsf{J}_{\mathbf{g}}(\mathbf{x})^\top \mathbf{d} + \mathbf{g}(\mathbf{x}) \leq \mathbf{0} \end{cases}. \tag{A.24}$$

Il existe des méthodes spécialisées dans la résolution de tels problèmes d'optimisation quadratique. Elles ont l'avantage d'être plus rapide que de calculer l'incrément à partir de (A.23), et sont aussi plus stables numériquement.

A.4 Robustesse

Les données manipulées par les algorithmes présentés ici ne sont pas toujours exactes. Différentes causes peuvent provoquer cela :

- Hypothèse du modèle non respectée ou trop grossière pour une partie des données ;
- Le bruit ou l'imprécision sur les données inhérents à toute mesure ;
- Les données tout simplement erronées (erreurs de mise en correspondance par exemple).

Cela a pour conséquence de fausser les résultats, parfois légèrement, mais bien souvent de manière très prononcée. Ces différentes causes peuvent être circonscrites ou amoindries en écartant les données qui sont incorrectes lors de l'optimisation. La séparation des données en deux ensembles, corrects et incorrects, peut être réalisée automatiquement de plusieurs manières à l'aide de méthodes statistiques.

Nous avons utilisé une de ces méthodes à plusieurs reprises, de manière directe ou indirecte puisque la boite à outils que nous avons utilisé pour l'étalonnage intègre, à travers cette méthode, la robustesse . Cette dernière, nommée RANSAC pour «RANdom SAmple Consensus» (Fischler and Bolles, 1981; Hartley and Zisserman, 2003; Forsyth and Ponce, 2003), procède de manière itérative pour éliminer les données qui sont aberrantes :

1. Tout d'abord, une partie des données de la taille n minimum nécessaire pour effectuer l'estimation est sélectionnée aléatoirement.
2. Une estimation des paramètres du modèle est réalisée à partir de ces données.
3. Le reste des données est filtré en fonction de leur erreur avec le modèle estimé : les données dont l'erreur est trop importante sont éliminées, ce sont des données erronées.
4. Une nouvelle estimation des paramètres est réalisée avec les données correctes qui ont été conservées à l'étape précédente.
5. L'erreur globale de ce nouveau modèle estimé est calculée puis comparée à la valeur de celle du meilleur modèle estimé jusqu'à présent : si elle est moindre nous considèrons que ce nouveau modèle est le meilleur.

L'ensemble de ces étapes est répété un nombre de fois suffisant pour que statistiquement la meilleure solution soit trouvée avec une certaine probabilité. Plus nous cherchons à ce que cette probabilité soit grande, et plus il est nécessaire de faire d'itérations. De même, plus il y a de données aberrantes et plus le nombre d'itération pour trouver la meilleure estimation tend à augmenter. Connaissant la probabilité w que le tirage aléatoire d'un point des données fasse partie des données correctes, il est possible de calculer une borne k théorique au nombre d'itération nécessaire pour obtenir un résultat correct avec une certaine probabilité p :

$$k = \frac{\log(1-p)}{\log(1-w^n)}. \tag{A.25}$$

Cependant la probabilité w de sélectionner un point appartenant aux données correctes n'étant pas toujours connues à l'avance, il existe des méthodes automatiques pour calculer le nombre d'itérations en même temps que l'estimation est réalisée (Hartley and Zisserman, 2003; Forsyth and Ponce, 2003).

Ces méthodes calculent à chaque itération une nouvelle borne en tenant compte de la proportion de données erronées avec le modèle estimé à l'itération en cours.

Il est possible d'arrêter prématurément la boucle lors de la cinquième étape si nous considèrons que, en dessous d'un certain seuil pour l'erreur globale, le nouveau modèle est suffisamment bon pour être celui qu'on recherchait. Ce seuil doit être ajusté manuellement en fonction du problème et de la qualité attendue des données. Il existe un autre seuil qui nécessite également un réglage, celui qui sert à séparer les données en données correctes et erronées lors de la troisième étape. Ces seuils sont bien souvent déterminés de manière empirique bien qu'il existe quelques méthodes destinées à les calculer automatiquement.

ANNEXE B

DÉCOMPOSITIONS MATRICIELLE

Les décompositions matricielles sont des méthodes mathématiques pour exprimer une matrice donnée en un produit de plusieurs autres matrices. Ces dernières peuvent avoir des propriétés particulières comme l'orthogonalité par exemple. Il est parfois nécessaire pour cela d'imposer certaines conditions sur la matrice à décomposer, comme le fait d'être carrée. Les propriétés des principales méthodes de décomposition sont présentées dans cette annexe.

B.1 Décomposition LU et de Cholesky

La décomposition LU est une décomposition d'une matrice carrée inversible A de taille $n \times n$ sous la forme $\mathsf{A} = \mathsf{LU}$ avec L qui est une matrice triangulaire inférieure de taille $n \times n$ et U qui est une matrice triangulaire supérieure de taille $n \times n$ également. Sous cette forme précise, la décomposition n'est possible que pour certaines matrices A : celles dont toutes les sous-matrices principales d'ordre 1 à $n - 1$ sont inversibles, ce qui implique que $\det(\mathsf{A}) \neq 0$.

Dans ce cas, le calcul de cette décomposition est relativement simple et est un cas particulier de l'élimination de Gauss-Jordan. Il consiste à calculer une suite de matrices triangulaires inférieures qui éliminent par produit avec A les éléments inférieurs à la diagonale, colonne par colonne, résultant ainsi en la matrice triangulaire supérieure recherchée. Le produit de toutes les matrices triangulaires inférieures est la matrice triangulaire inférieure souhaitée.

Il existe une forme plus générale de cette décomposition qui peut s'appliquer quelque soit la matrice carrée inversible A. Elle consiste à ajouter une matrice de permutation P dans la décomposition qui devient alors $\mathsf{A} = \mathsf{PLU}$. En effet, lors du calcul des matrices inférieures, il est nécessaire que l'élément diagonal sur la colonne concernée soit non nul. Lorsque ce cas de figure se produit, il suffit alors de

permuter la ligne où se trouve cet élément diagonal nul avec une autre ligne où il est non nul [1], d'où la matrice de permutation.

Lorsque la matrice A est symétrique et définie positive, cette décomposition se simplifie en la décomposition suivante : $A = LL^\top$, qui est dite décomposition de Cholesky. Si nous imposons que les éléments diagonaux de la matrice soient tous positifs, cette décomposition est alors unique.

B.2 Décompositions QR, RQ, LQ et QL

La décomposition QR s'applique à n'importe quelle matrice A, qu'elle décompose sous la forme du produit d'une matrice orthogonale Q de même taille que A et d'une matrice triangulaire supérieure R : $A = QR$. L'ordre des deux matrices a son importance. Ainsi, il existe aussi la décomposition RQ où le produit est effectué dans l'autre sens : $A = RQ$. La matrice triangulaire supérieure peut aussi être remplacée par une matrice triangulaire inférieure L, donnant ainsi les décompositions LQ ($A = LQ$) et QL ($A = QL$).

Quelque soit la forme souhaitée, il existe trois méthodes de calcul de cette décomposition. Chacune de ces méthodes donnant un résultat différent, cette décomposition n'étant pas unique. La plus fréquemment utilisée fait appel à une suite de matrices orthogonales élémentaires nommées matrices de Householder. Chacune de ces matrices est calculée pour éliminer les éléments du mauvais côté de la diagonale d'une colonne donnée. Le produit de toutes ces matrices forme la matrice orthogonale, tandis que le produit de la transposée de cette matrice orthogonale par la matrice A est la matrice triangulaire recherchée.

B.3 Décomposition en valeurs singulières

La décomposition en valeurs singulières (Golub and Reinsch, 1970), souvent nommée SVD («Singular Value Decomposition»), est une décomposition fondamentale en algèbre. Elle a de nombreuses propriétés très utiles. Bien qu'elle soit définie aussi bien pour les matrices à coefficients réels que complexes, seul le premier cas sera développé.

B.3.1 Théorème

Pour toute matrice à coefficients réels A de taille $n \times m$, il existe une décomposition de la forme $A = USV^\top$ et telle que :

1. Il existe nécessairement une telle ligne étant donné que la matrice est inversible.

- U soit une matrice orthonormale de taille $m \times m$;
- S soit une matrice diagonale de taille $m \times n$ dont tous les coefficients sont positifs ou nuls ;
- V soit une matrice orthonormale de taille $n \times n$.

Les valeurs contenues dans S sont appelées valeurs singulières de A. Il existe $\min(n, m)$ valeurs singulières. Lorsqu'elles sont triées par ordre décroissant, la matrice S est unique. Les colonnes des matrices U et V contiennent les vecteurs singuliers de A à gauche et à droite formant respectivement des bases orthonormées de $\mathbb{R}^m$ et de $\mathbb{R}^n$. Les vecteurs singuliers de A sont uniques à leur signe près. La i-ème valeur singulière s_i est associée à la i-ème colonne de U et V par les relations $\mathsf{A}\mathbf{v}_i = s_i\mathbf{u}_i$ et $\mathsf{A}^\top\mathbf{u}_i = s_i\mathbf{v}_i$.

Il existe plusieurs méthodes de calculs de la SVD : l'algorithme de Golub-Reinsch, l'orthogonalisation de Jacobi ou encore l'utilisation de matrices de Householder comme pour la décomposition QR. Chacune présente des avantages et des inconvénients. Le calcul complet de la SVD nécessite $4m^2n + 8mn^2 + 9n^3$ opérations en virgules flottantes. Lorsque la matrice U n'est pas nécessaire, il est bien plus rapide de calculer uniquement S et V, car le coût en opérations à virgules flottantes est alors de $4mn^2 + 8n^3$.

B.3.2 Propriétés

De par la relation entre vecteurs singuliers et valeurs singulières, il est possible de remarquer que lorsque la i-ème valeur singulière est nulle, alors pour le i-ème vecteur singulier à droite $\mathsf{A}\mathbf{v}_i = \mathbf{0}$. Par conséquent, les vecteurs singuliers à droite correspondant à une valeur singulière nulle appartiennent au noyau de A, et étant donné qu'ils sont orthogonaux, ils en forment une base.

De la même manière, les vecteurs singuliers à gauche correspondant à des valeurs singulières non-nulles forment une base de l'image de A. Le rang de A est donc égal au nombre de valeurs singulières non-nulles.

En utilisant la SVD de A, il est possible d'en calculer une pseudo-inverse : $\mathsf{A}^+ = \mathsf{V}S^+\mathsf{U}^\top$, où S^+ est la matrice S dont chaque valeur singulière non-nulle a été remplacée par son inverse. Cette pseudo-inverse est souvent référencée comme la pseudo-inverse de Moore-Penrose, bien qu'il existe d'autres méthodes de calcul de cette dernière. Elle possède les propriétés suivantes :

- $\mathsf{A}\mathsf{A}^+\mathsf{A} = \mathsf{A}$ et $\mathsf{A}^+\mathsf{A}\mathsf{A}^+ = \mathsf{A}^+$;
- $\mathsf{A}\mathsf{A}^+$ et $\mathsf{A}^+\mathsf{A}$ sont hermitiennes.

ANNEXE C

ROTATIONS TRIDIMENSIONNELLES

C.1 Caractérisation et propriétés

Dans un espace vectoriel euclidien orienté de dimension trois, la transformation consistant à faire tourner un vecteur d'un certain angle autour d'un axe donné est appelée rotation vectorielle. Du point de vue de l'algèbre linéaire, cette transformation peut s'exprimer sous la forme d'une matrice 3×3 dite de rotation. Exprimée dans une base orthonormée directe de l'espace vectoriel, cette matrice est orthogonale directe. En la notant R, elle vérifie donc les équations données par

$$\begin{cases} \mathsf{R}\mathsf{R}^\top = \mathsf{R}^\top\mathsf{R} = \mathtt{I}_3 \\ |\mathsf{R}| = \det(\mathsf{R}) = 1. \end{cases} \tag{C.1}$$

L'ensemble des matrices de rotation muni de la multiplication matricielle habituelle forme un groupe, appelé groupe spécial orthogonal et noté classiquement $\mathcal{SO}_3(\mathbb{R})$. Les propriétés suivantes sont donc vérifiées par les matrices de rotations :

- Le produit de deux matrices de rotation est une matrice de rotation ;
- La matrice identité $\mathtt{I}_3$ est l'élément neutre du groupe ;
- Toute matrice de rotation possède une inverse, il s'agit de sa transposée.

Bien que les matrices de rotation aient neuf entrées, le nombre de degrés de liberté est de trois en raison des contraintes de (C.1). La paramétrisation directe d'une matrice de rotation par neuf paramètres nécessite donc d'imposer ces contraintes, ce qui peut parfois s'avérer complexe.

C.2 Angles d'Euler

Le produit de matrices de rotation étant une matrice de rotation, il est possible de former une paramétrisation minimale des matrices de rotation en considérant comme paramètres les angles de rotations autour de trois axes distincts donnés. Il est assez intuitif de choisir les trois axes comme étant ceux de la base canonique de l'espace vectoriel. L'expression des trois matrices de rotations est alors très simple puisque ce sont des rotations dans le plan normal à chaque axe :

$$\mathsf{R}_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) \\ 0 & \sin(\alpha) & \cos(\alpha) \end{bmatrix} \tag{C.2}$$

$$\mathsf{R}_y(\beta) = \begin{bmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{bmatrix} \tag{C.3}$$

$$\mathsf{R}_z(\gamma) = \begin{bmatrix} \cos(\gamma) & -\sin(\gamma) & 0 \\ \sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{C.4}$$

Il faut néanmoins faire attention à l'ordre dans lequel sont appliquées ces trois rotations.

Bien que moins intuitif, il existe un autre choix d'axes qui permet de paramétrer toute rotation par le produit de trois matrices de rotation simple. Les trois angles correspondants à ce choix sont appelés angles d'Euler et se décomposent selon

- La précession d'angle ψ autour de $\vec{z}$;
- La nutation d'angle θ autour de $\vec{x}$;
- La rotation propre d'angle φ autour de $\vec{z}$.

La matrice de rotation correspondante est donnée par le produit matriciel

$$\mathsf{R}_z(\varphi)\mathsf{R}_x(\theta)\mathsf{R}_z(\psi) = \begin{bmatrix} \cos(\varphi)\cos(\psi) - \sin(\varphi)\cos(\theta)\sin(\psi) & -\cos(\varphi)\sin(\psi) - \sin(\varphi)\cos(\theta)\cos(\psi) & \sin(\varphi)\sin(\theta) \\ \sin(\varphi)\cos(\psi) + \cos(\varphi)\cos(\theta)\sin(\psi) & -\sin(\varphi)\sin(\psi) + \cos(\varphi)\cos(\theta)\cos(\psi) & -\cos(\varphi)\sin(\theta) \\ \sin(\theta)\sin(\psi) & \sin(\theta)\cos(\psi) & \cos(\theta) \end{bmatrix}. \tag{C.5}$$

Sa dérivée par rapport à ψ est

$$\frac{d\mathsf{R}_z(\varphi)\mathsf{R}_x(\theta)\mathsf{R}_z(\psi)}{d\psi} = \begin{bmatrix} -\cos(\varphi)\sin(\psi) - \sin(\varphi)\cos(\theta)\cos(\psi) & -\cos(\varphi)\cos(\psi) + \sin(\varphi)\cos(\theta)\sin(\psi) & 0 \\ -\sin(\varphi)\sin(\psi) + \cos(\varphi)\cos(\theta)\cos(\psi) & -\sin(\varphi)\cos(\psi) - \cos(\varphi)\cos(\theta)\sin(\psi) & 0 \\ \sin(\theta)\cos(\psi) & -\sin(\theta)\sin(\psi) & 0 \end{bmatrix}. \quad \text{(C.6)}$$

Sa dérivée par rapport à θ est

$$\frac{d\mathsf{R}_z(\varphi)\mathsf{R}_x(\theta)\mathsf{R}_z(\psi)}{d\theta} = \begin{bmatrix} \sin(\varphi)\sin(\theta)\sin(\psi) & \sin(\varphi)\sin(\theta)\cos(\psi) & \sin(\varphi)\cos(\theta) \\ -\cos(\varphi)\sin(\theta)\sin(\psi) & -\cos(\varphi)\sin(\theta)\cos(\psi) & -\cos(\varphi)\cos(\theta) \\ \cos(\theta)\sin(\psi) & \cos(\theta)\cos(\psi) & -\sin(\theta) \end{bmatrix}. \quad \text{(C.7)}$$

Sa dérivée par rapport à φ est

$$\frac{d\mathsf{R}_z(\varphi)\mathsf{R}_x(\theta)\mathsf{R}_z(\psi)}{d\varphi} = \begin{bmatrix} -\sin(\varphi)\cos(\psi) - \cos(\varphi)\cos(\theta)\sin(\psi) & \sin(\varphi)\sin(\psi) - \cos(\varphi)\cos(\theta)\cos(\psi) & \cos(\varphi)\sin(\theta) \\ \cos(\varphi)\cos(\psi) - \sin(\varphi)\cos(\theta)\sin(\psi) & -\cos(\varphi)\sin(\psi) - \sin(\varphi)\cos(\theta)\cos(\psi) & \sin(\varphi)\sin(\theta) \\ 0 & 0 & 0 \end{bmatrix}. \quad \text{(C.8)}$$

C.3 Quaternions unitaires

Les quaternions sont une construction mathématique généralisant les nombres complexes. Ce sont des quadruplets de nombres réels qui forment une algèbre à division contenant l'addition vectorielle habituelle et une multiplication spécifique. Cette algèbre est classiquement notée $\mathbb{H}$.

Notons $\mathbf{q} \in \mathbb{H}$ un quaternion, il s'écrit alors comme $\mathbf{q} = [q_1, q_2, q_3, q_4]^\top \in \mathbb{R}^4$. En supposant que ce quaternion est unitaire, autrement dit que $q_1^2 + q_2^2 + q_3^2 + q_4^2 = 1$, alors ce quaternion permet de représenter une rotation tridimensionnelle. Elle est donnée par

$$\mathsf{R}(\mathbf{q}) = \begin{bmatrix} q_1^2 + q_2^2 - q_3^2 - q_4^2 & 2q_2q_3 - 2q_1q_4 & 2q_1q_3 + 2q_2q_4 \\ 2q_1q_4 + 2q_2q_3 & q_1^2 - q_2^2 + q_3^2 - q_4^2 & 2q_3q_4 - 2q_1q_2 \\ 2q_2q_4 - 2q_1q_3 & 2q_1q_2 + 2q_3q_4 & q_1^2 - q_2^2 - q_3^2 + q_4^2 \end{bmatrix}. \quad \text{(C.9)}$$

Les dérivées de cette matrice par rapport aux éléments de $\mathbf{q}$ sont facilement calculées :

$$\frac{d\mathsf{R}(\mathbf{q})}{dq_1} = \begin{bmatrix} 2q_1 & -2q_4 & 2q_3 \\ 2q_4 & 2q_1 & -2q_2 \\ -2q_3 & 2q_2 & 2q_1 \end{bmatrix}, \tag{C.10}$$

$$\frac{d\mathsf{R}(\mathbf{q})}{dq_2} = \begin{bmatrix} 2q_2 & 2q_3 & 2q_4 \\ 2q_3 & -2q_2 & -2q_1 \\ 2q_4 & 2q_1 & -2q_2 \end{bmatrix}, \tag{C.11}$$

$$\frac{d\mathsf{R}(\mathbf{q})}{dq_3} = \begin{bmatrix} -2q_3 & 2q_2 & 2q_1 \\ 2q_2 & 2q_3 & 2q_4 \\ -2q_1 & 2q_4 & -2q_3 \end{bmatrix}, \tag{C.12}$$

$$\frac{d\mathsf{R}(\mathbf{q})}{dq_4} = \begin{bmatrix} -2q_4 & -2q_1 & 2q_2 \\ 2q_1 & -2q_4 & 2q_3 \\ 2q_2 & 2q_3 & 2q_4 \end{bmatrix}. \tag{C.13}$$

Avec cette représentation, il est possible de décomposer le quaternion en un scalaire (q_1 dans cette paramétrisation) et un vecteur tridimensionnel ($[q_2, q_3, q_4]^\top$). La rotation représentée par ce quaternion est alors une rotation d'angle $\theta = 2\cos^{-1}(q_1)$ autour de l'axe de support du vecteur. Un quaternion correspondant à la rotation inverse $\mathsf{R}^{-1} = \mathsf{R}^\top$ peut alors être obtenu simplement en changeant seulement le signe de q_1. Le même résultat est obtenu en changeant uniquement le signe du vecteur. Les quaternions forment en fait un revêtement double de l'ensemble des rotations $\mathcal{SO}_3(\mathbb{R})$: $\mathbf{q}$ et $-\mathbf{q}$ correspondent à la même rotation tridimensionnelle.

Ne considérer que les quaternions tels que $q_1 \in \mathbb{R}^+$ permet d'obtenir un revêtement simple des rotations en ne choisissant qu'un des deux quaternions. Cela revient à imposer que l'angle de la rotation correspondante soit dans $[-\pi; \pi]$ au lieu de $[-2\pi; 2\pi]$. Il est aussi possible d'imposer que l'axe de la rotation, donné par le vecteur $[q_2, q_3, q_4]^\top$, se trouve au dessus du plan tridimensionnel défini par $q_2 + q_3 + q_4 = 0$. Une dernière possibilité triviale pour obtenir un revêtement simple est de ne considérer que les quaternions tels que $q_1 + q_2 + q_3 + q_4 \in \mathbb{R}^+$. Cela revient à diviser l'hypersphère unitaire par l'hyperplan d'équation $q_1 + q_2 + q_3 + q_4 = 0$.

C.4 Interpolation de rotations

L'interpolation de rotations, qui permet de passer d'une rotation à une autre progressivement, est une problématique importante en imagerie. Il existe de nombreuses méthodes pour calculer les rota-

tions intermédiaires, chacune ayant des avantages et des inconvénients. Bien que des méthodes basées sur la paramétrisation directe des rotations aient été proposées, elles sont complexes et peu efficaces. En utilisant la paramétrisation par des quaternions unitaires des rotations, deux méthodes sont classiquement utilisées.

La méthode la plus intuitive, et aussi incontestablement la plus efficace, est de réaliser une interpolation linéaire entre les deux quaternions puis de normaliser le quaternion résultant. Cette approche tire partie du fait que la projection centrale d'un quaternion sur la sphère unité donne le quaternion correspondant à la rotation la plus proche. Elle présente cependant un désavantage qui réside dans le fait que la vitesse à laquelle on passe de la première rotation à la deuxième n'est pas uniforme : aux alentours des rotations initiale et finale la progression se fait lentement, alors qu'elle se fait rapidement proche du milieu.

L'autre méthode est nommée SLERP (pour «Spherical Linear intERPolation») (Shoemake, 1985) et bien que moins efficace, elle a l'avantage de se faire à vitesse constante. Le principe est de calculer l'angle α de la rotation entre les deux axes des rotations à interpoler puis de se déplacer progressivement de l'une à l'autre en effectuant une rotation proportionnelle à cet angle. Le produit scalaire des deux quaternions donne $\cos(\alpha)$, ce qui permet de calculer $\theta = t\alpha$ où t est le coefficient de l'interpolation. Le quaternion de la rotation finale est alors donné par

$$\mathbf{q}_t = \frac{\sin(\alpha - t\alpha)\mathbf{q}_0 + \sin(t\alpha)\mathbf{q}_1}{\sin(\alpha)}, \tag{C.14}$$

où $\mathbf{q}_0$ est le quaternion de la rotation initiale et $\mathbf{q}_1$ celui de la rotation finale. Il faut noter deux petits détails :

- tout d'abord si les deux rotations ont des quaternions colinéaires (ou quasi colinéaires) alors $\sin(\alpha)$ est nul (ou proche de l'être), dans ce cas il vaut mieux utiliser la méthode précédente ;
- ensuite lorsque $\cos(\alpha)$ est négatif, cette méthode réalise une interpolation qui passe par le plus long chemin entre les deux rotations, dans ce cas il faut prendre l'opposé d'un des deux quaternions (qui représente la même rotation) pour avoir le plus court chemin.

ANNEXE D

RECONSTRUCTION STÉRÉOSCOPIQUE

À l'instar de nos deux yeux qui nous permettent de voir en 3D, à partir d'une paire de caméras fixées l'une par rapport à l'autre, et dont les champs se recouvrent, il est possible de reconstruire la structure tridimensionnelle d'une scène ou des objets. Cette opération s'appelle la reconstruction stéréoscopique. Elle procède par minimisation de l'erreur de reprojection dans les deux images, ce qui aboutit à la résolution d'un système linéaire sur les coordonnées 3D des points de la scène qui sont mis en correspondance sur les deux images. La paire de caméras, dite elle aussi stéréoscopique, doit être étalonnée préalablement pour obtenir une reconstruction qui soit correcte et à l'échelle. Dans cette annexe, les exposants sur les projections et les matrices de projections indiquent l'image considérée.

D.1 Modèle de projection

Pour chacune des deux caméras de la paire stéréo, la projection est modélisée de la même manière que dans la section II.2. Il y a donc deux matrices de projections M^1 et M^2, exprimées dans le même repère monde. Un point de la scène est projeté dans chacune des deux images selon

$$\begin{cases} \tilde{\mathbf{m}}_i^1 = \mathsf{M}^1 \tilde{\mathbf{p}}_i \\ \tilde{\mathbf{m}}_i^2 = \mathsf{M}^2 \tilde{\mathbf{p}}_i. \end{cases} \tag{D.1}$$

Ceci est illustré dans la figure D.1 Un exemple de vues issues d'une paire de caméras stéréo est donné dans la figure D.2.

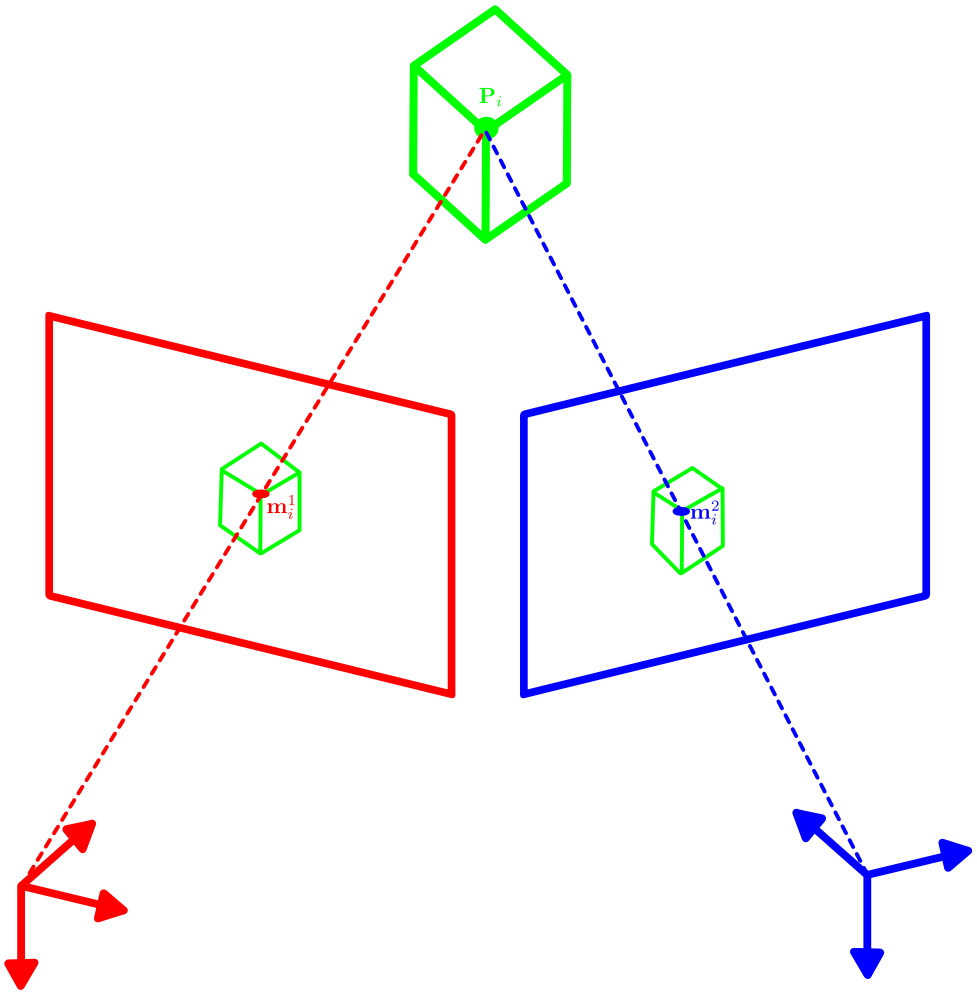
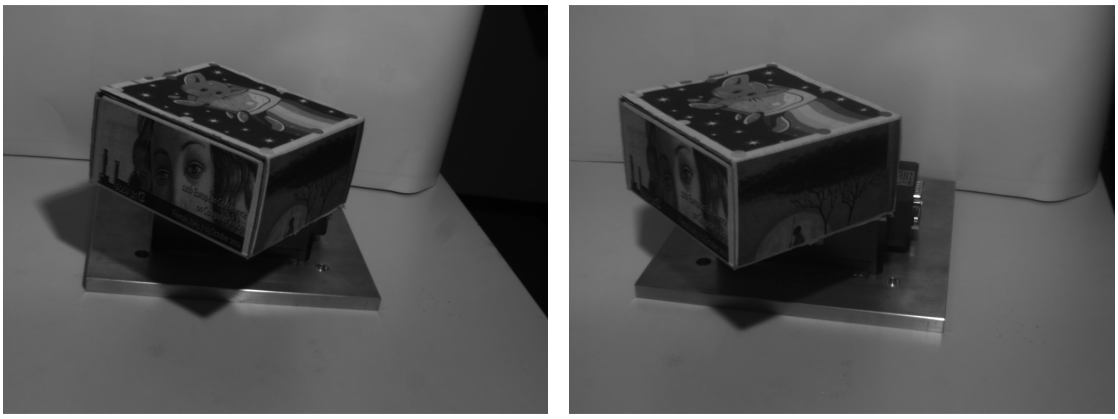


FIGURE D.1 – Modélisation de la projection stéréo : deux caméras fixes l'une par rapport à l'autre observent un objet. Les deux matrices de projection M^1 et M^2 sont supposées connues. Un point de la scène est projeté dans chacune des deux images selon le modèle de projection perspective de la section II.2.



(a) Image de la caméra gauche. (b) Image de la caméra droite.

FIGURE D.2 – Exemples d'images issues d'une paire de caméras stéréo.

D.2 Étalonnage d'une paire stéréoscopique de caméras

Dans le cas général, l'étalonnage d'une paire stéréoscopique de caméras est une tâche complexe. Néanmoins lorsque les deux caméras de la paire stéréoscopique ont été préalablement étalonnées avec plusieurs vues d'un même objet, auquel est lié le repère monde, il ne reste qu'à retrouver la position d'une des deux caméras par rapport à l'autre. Pour cela, considérons que les deux caméras sont étalonnées pour chaque vue selon

$$\begin{cases} \mathsf{M}_j^1 = \mathsf{K}^1 \left[\begin{array}{c} \mathsf{R}_j^1 \mathbf{t}_j^1 \end{array} \right] \\ \mathsf{M}_j^2 = \mathsf{K}^2 \left[\begin{array}{c} \mathsf{R}_j^2 \mathbf{t}_j^2 \end{array} \right] \end{cases}, \tag{D.2}$$

où l'indice j indique la vue considérée. Pour la vue j, la rotation entre les deux caméras est donnée par

$$\mathsf{R}_j = \mathsf{R}_j^1 \mathsf{R}_j^{2^\top}, \tag{D.3}$$

et la translation entre les deux caméras est alors

$$\mathbf{t}_j = \mathbf{t}_j^1 - \mathsf{R}_j \mathbf{t}_j^2. \tag{D.4}$$

Une fois ces valeurs calculées pour chacune des vues, la moyenne des paramètres est réalisée pour obtenir une valeur initiale globale. Une optimisation de l'erreur de reprojection dans l'ensemble des vues permet alors de raffiner ces paramètres. C'est ce qui est réalisé dans la boîte-à-outils (Bouguet,

2010).

D.3 Reconstruction tridimensionnelle

Du modèle de projection (D.1), le facteur d'échelle homogène peut être éliminé en utilisant le produit vectoriel :

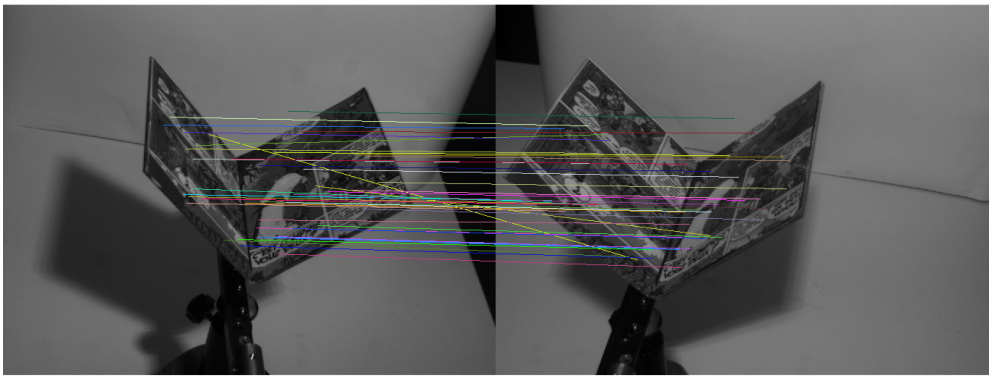
$$\begin{cases} \left[\tilde{\mathbf{m}}_i^1\right]_\wedge \mathsf{M}^1 \tilde{\mathbf{p}}_i = \mathbf{0} \\ \left[\tilde{\mathbf{m}}_i^2\right]_\wedge \mathsf{M}^2 \tilde{\mathbf{p}}_i = \mathbf{0}. \end{cases} \tag{D.5}$$

Sur les trois équations fournies par chaque image, seulement deux sont linéairement indépendantes, la troisième peut donc être éliminée

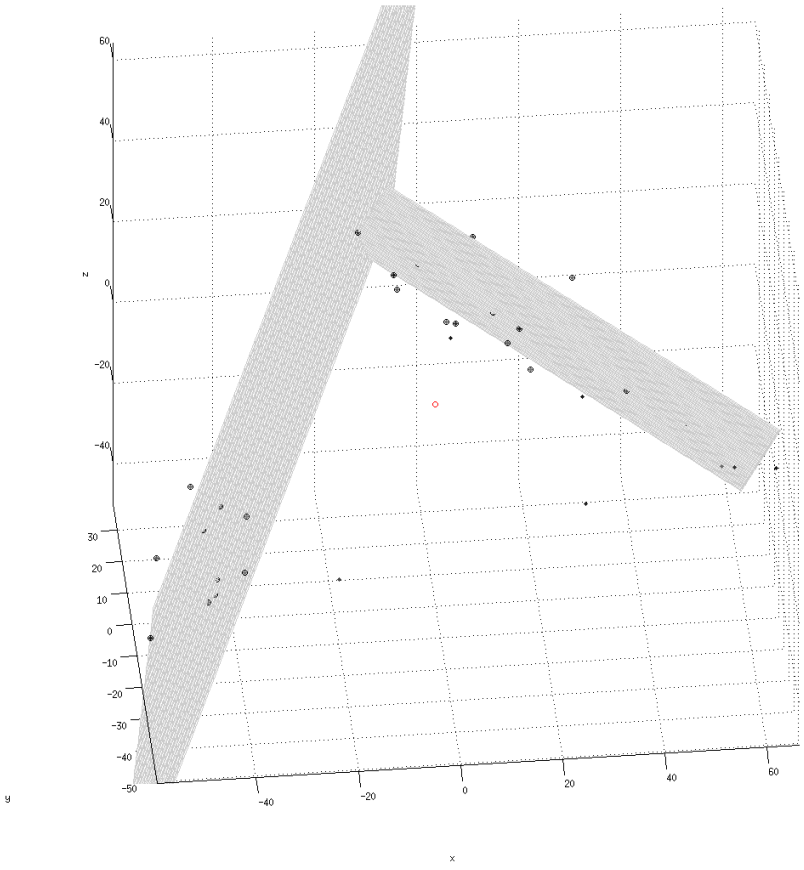
$$\begin{cases} \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}}_{=\mathsf{S}} \left[\tilde{\mathbf{m}}_i^1\right]_\wedge \mathsf{M}^1 \tilde{\mathbf{p}}_i = \mathbf{0} \\ \mathsf{S}\left[\tilde{\mathbf{m}}_i^2\right]_\wedge \mathsf{M}^2 \tilde{\mathbf{p}}_i = \mathbf{0} \end{cases}. \tag{D.6}$$

Il en résulte ainsi un système de quatre équations linéaires par rapport aux trois inconnues de $\mathbf{p}_i$. Ce système peut s'écrire sous forme matricielle $\mathsf{A}\tilde{\mathbf{p}}_i = \mathbf{0}$ et être résolu en utilisant la méthode décrite section A.1. Un exemple de reconstruction obtenue de cette manière est donné dans la figure D.3.

Une autre méthode pour estimer $\mathbf{p}_i$ existe et est décrite dans (Hartley and Zisserman, 2003). Elle est basée sur la minimisation de l'erreur de reprojection dans chacune des images sous la contrainte épipolaire. Ceci aboutit alors à la résolution d'une équation polynomiale de degré six pour trouver les minima de l'erreur de reprojection. Cela peut être réalisé de manière globale et non-itérative. Cette méthode a l'avantage de fournir un estimateur optimal si le bruit sur les projections est Gaussien, ce qui n'est pas le cas de la méthode présentée précédemment.



(a) Images de l'objet et correspondances.



(b) Reconstruction tridimensionnelle.

FIGURE D.3 – Exemple d'une reconstruction tridimensionnelle à partir d'une paire de caméras stéréo. Deux plans ont été ajustés au nuage de points obtenu après la reconstruction.

BIBLIOGRAPHIE

Ait-Aider, O., Andreff, N., Lavest, J.-M., and Martinet, P. (2006).
Simultaneous object pose and velocity computation using a single view from a rolling shutter camera.
In *ECCV*.

Ait-Aider, O., Bartoli, A., and Andreff, N. (2007).
Kinematics from lines in a single rolling shutter image.
In *CVPR*.

Ait-Aider, O. and Berry, F. (2009).
Structure and kinematics triangulation with a rolling shutter stereo rig.
In *ICCV*.

Ameller, M.-A., L., Q., and B., T. (2002).
Camera pose revisited : New linear algorithms.
In *RFIA*.

Anderson, E., Bai, Z., Bischof, C., Blackford, S., Demmel, J., Dongarra, J., Du Croz, J., Greenbaum, A., Hammarling, S., McKenney, A., and Sorensen, D. (1999).
LAPACK users' guide.
Society for Industrial and Applied Mathematics.

Axler, S. J. (1997).
Linear Algebra Done Right.
Springer-Verlag.

Baer, R. (1952).

Linear algebra and projective geometry.
Academic Press.

Bay, H., Tuytelaars, T., and Gool, L. V. (2006).
SURF : Speeded up robust features.
In *ECCV*.

Bonesky, T. (2009).
Morozov's discrepancy principle and tikhonov-type functionals.
Inverse Problems.

Borchers, B. (1999).
Csdp, a c library for semidefinite programming.
Optimization Methods and Software.

Borchers, B. and Young, J. G. (2007).
Implementation of a primal–dual method for sdp on a shared memory parallel architecture.
Computational Optimization and Applications.

Bouguet, J.-Y. (2010).
Camera calibration toolbox for matlab.
http://www.vision.caltech.edu/bouguetj/calib-doc/index.html.

Bradley, D., Atcheson, B., Ihrke, I., and Heidrich, W. (2009).
Synchronization and rolling shutter compensation for consumer video camera arrays.
In *PROCAMS*.

Broyden, C. G. (1970).
The convergence of a class of double-rank minimization algorithms.
Journal of the Institute of Mathematics and Its Applications.

Brunet, F., Bartoli, A., Malgouyres, R., and Navab, N. (2008).
L-tangent norm : A low computational cost criterion for choosing regularization weights and its use for range surface reconstruction.
In *3DPVT*.

Byröd, M., Josephson, K., and Aström, K. (2009).
Fast and stable polynomial equation solving and its application to computer vision.
IJCV.

Cho, W.-H. and Hong, K.-S. (2007).
Affine motion based cmos distortion analysis and cmos digital image stabilization.
IEEE Trans. on Consumer Electronics.

Cho, W.-H. and Hong, K.-S. (2008).
A fast cis still image stabilization method without parallax and moving object problems.
IEEE Trans. on Consumer Electronics.

Cho, W.-H., Kim, D.-W., and Hong, K.-S. (2007).
Cmos digital image stabilization.
IEEE Trans. on Consumer Electronics.

Cho, W.-H. and Kim, T.-C. (2012).
Cis video panoramic image.
In *ISCE*.

Chun, J.-B., Jung, H., and Kyung, C.-M. (2008).
Suppressing rolling-shutter distortion of cmos image sensors by motion vector detection.
IEEE Trans. on Consumer Electronics.

Coxeter, H. S. M. (2003).
Projective Geometry.
Springer.

Cruset, J. (1966).
Leçons d'optique appliquée et de photographie.
École nationale des sciences géographiques.

Dahmouche, R., Ait-Aider, O., Andreff, N., and Mezouar, Y. (2008).
High-speed pose and velocity measurement from vision.
In *ICRA*.

Dahmouche, R., Andreff, N., Mezouar, Y., and Martinet, P. (2009).
3D pose and velocity visual tracking based on sequential region of interest acquisition.
In *IROS*.

D'Angelo, E. P. A. and Vandergheynst, P. (2011).
Method to compensate the effect of the rolling shutter effect.

Dementhon, D. and Davis, L. (1995).
Model-based object pose in 25 lines of code.

IJCV.

Dongarra, J. J., Du Croz, J., Duff, I. S., and Hammarling, s. (1990).
A set of level 3 basic linear algebra subprograms.
ACM Trans. Math. Soft.

Fan, Y., Sarkar, S., and Lasdon, L. (1988).
Experiments with successive quadratic programming algorithms.
Journal of Optimization Theory and Applications.

Fischler, M. A. and Bolles, R. C. (1981).
Random sample consensus : A paradigm for model fitting with applications to image analysis and automated cartography.
Comm. of the ACM.

Fletcher, R. (1970).
A new approach to variable metric algorithms.
Computer Journal.

Fletcher, R. (1987).
Practical Methods of Optimization.
John Wiley and Sons.

Forssén, P.-E. and Ringaby, E. (2010).
Rectifying rolling shutter video from hand-held devices.
In *CVPR*.

Forsyth, D. and Ponce, J. (2003).
Computer Vision a Modern Approach.
Prentice Hall.

Fryer, J. and Brown, D. (1986).
Lens distortion for close-range photogrammetry.
Photogrammetric Engineering and Remote Sensing.

Goldfarb, D. (1970).
A family of variable metric updates derived by variational means.
Mathematics of Computation.

Golub, G. and Reinsch, C. (1970).
Singular value decomposition and least squares solutions.

Numerische Mathematik.

Hansen, P. (1992).
Analysis of discrete ill-posed problems by means of the l-curve.
SIAM.

Hansen, P. (2005).
The l-curve and its use in the numerical treatment of inverse problems.
Technical report, Technical University of Denmark.

Hartley, R. and Gupta, R. (1994).
Linear pushbroom cameras.
In *ECCV*.

Hartley, R. and Zisserman, A. (2003).
Multiple View Geometry in Computer Vision.
Cambridge Univ. Press.

Hedborg, J., Forssén, P.-E., Felsberg, M., and Ringaby, E. (2012).
Rolling shutter bundle adjustment.
In *CVPR*.

Hedborg, J., Ringaby, E., Forssén, P.-E., and Felsberg, M. (2011).
Structure and motion estimation from rolling shutter video.
In *IWMV*.

Hedgecoe, J. (2009).
Le nouveau manuel de la photographie.
Pearson.

Heflin, B., Scheirer, W., and Boult, T. E. (2010).
Correcting rolling-shutter distortion of cmos sensors using facial feature detection.
In *BTAS*.

Heikkilä and Silvén (1997).
A four-step camera calibration procedure with implicit image correction.
In *CVPR*.

Henrion, D., Lasserre, J., and Loefberg, J. (2009).
Gloptipoly 3 : moments, optimization and semidefinite programming.
Optimization Methods and Software.

Hong, W., Wei, D., and Batur, A. U. (2012).
Video stabilization and reduction of rolling shutter distortion.

Hwang, J. H. (2003).
Cmos image sensor distortion compensation method.

Im, J.-A., Kim, D.-W., and Hong, K.-S. (2006).
Digital video stabilization algorithm for CMOS image sensor.

Kieffer, R. (1985).
Le point sur la netteté.
Nikon-News.

Klein, G. and Murray, D. (2009).
Parallel tracking and mapping on a camera phone.
In *ISMAR*.

Kohavi, R. (1995).
A study of cross-validation and bootstrap for accuracy estimation and model selection.
In *International Joint Conference on Artificial Intelligence*.

Kukelova, Z., Bujnak, M., and Pajdla, T. (2008).
Automatic generator of minimal problem solvers.
In *ECCV*.

Laroche, E. and Kagami, S. (2009).
Dynamical models for position measurement with global shutter and rolling shutter cameras.
In *IROS*.

Lasserre, J. (2001).
Global optimization with polynomials and the problem of moments.
SIAM.

Lasserre, J. (2008).
A semidefinite programming approach to the generalized problem of moments.
Mathematical Programming.

Lasserre, J. (2009).
Moments and sums of squares for polynomial optimization and related problems.
Journal of Global Optimization.

Laurent, M. (2009).
Sums of squares, moment matrices and optimization over polynomials.
IMA Volumes in Mathematics and its Applications.

Leon, S. J. (2006).
Linear Algebra With Applications.
Pearson Prentice Hall.

Lepetit, V., Moreno-Noguer, F., and Fua, P. (2009).
Epnp : An accurate O(n) solution to the PnP problem.
IJCV.

Levenberg, K. (1944).
A method for the solution of certain non-linear problems in least squares.
Quarterly of Applied Mathematics.

Liang, C.-K., Chang, L.-W., and Chen, H. H. (2008).
Analysis and compensation of rolling shutter effect.
IEEE Trans. on Image Processing.

Lowe, D. G. (2004).
Distinctive image features from scale-invariant keypoints.
IJCV.

Lu, S., Pereverzev, S., Shao, Y., and Tautenhahn, U. (2010).
On the generalized discrepancy principle for tikhonov regularization in hilbert scales.
Integral Equations Applied.

Lucas, B. and Kanade, T. (1981).
An iterative image registration technique with an application to stereo vision.
IJCAI.

Luxorion (2013).
La photographie numérique.
http://www.astrosurf.com/luxorion/photo-numerique.htm.

Magerand, L. and Bartoli, A. (2010).
A generic rolling shutter camera model and its application to dynamic pose estimation.
In *3DPVT.*

Magerand, L., Bartoli, A., Ait-Aider, O., and Pizarro, D. (2012).
Global optimization of object pose and motion from a single rolling shutter image with automatic 2d-3d matching.
In *ECCV*.

Marquardt, D. W. (1963).
An algorithm for least-squares estimation of nonlinear parameters.
SIAM.

Meingast, M., Geyer, C., and Sastry, S. (2005).
Geometric models of rolling-shutter cameras.
CoRR.

Moré, J. (1977).
The levenberg-marquardt algorithm : Implementation and theory.
Numerical Analysis.

Moussa, A. and Ponsonnet, P. (1975).
Cours de Physique, Optique.
André Desvigne.

Nicklin, S. P., Fisher, R. D., and Middleton, R. H. (2007).
Rolling shutter image compensation.
In *RoboCup 2006*.

Nocedal, J. and Wright, S. J. (2006).
Numerical Optimization.
Springer.

OpenMP (2000).
Openmp website.
http://openmp.org/.

Pajdla, T. (2002).
Geometry of two-slit camera.
Technical report, Czech Technical University.

Paragios, N., Chen, Y., and Faugeras, O. (2006).
Handbook of Mathematical Models in Computer Vision.
Springer.

Parrilo, P. (2003).
Semidefinite programming relaxations for semialgebraic problems.
Mathematical Programming.

Photographie (2013).
Photographie.
http://fr.wikibooks.org/wiki/Photographie.

Powell, M. (1983).
Variable metric methods for constrained optimization.
Mathematical Programming.

Prajna, S., Papachristodoulou, A., Seiler, P., and Parrilo, P. A. (2004).
SOSTOOLS : Sum of squares optimization toolbox for MATLAB.

ProDAD (2010).
Mercalli v2.
http://www.prodad.com.

Pérez, J.-P. (2004).
Optique : Fondements et applications.
Dunod.

Quan, L. (1999).
Linear n-point camera pose determination.
IEEE Trans. on Pattern Analysis and Machine Intelligence.

Ringaby, E. and Forssén, P.-E. (2011).
Scan rectification for structured light range sensors with rolling shutters.
In *ICCV*.

Ringaby, E. and Forssén, P.-E. (2012).
Efficient video rectification and stabilisation for cell-phones.
International Journal of Computer Vision.

Rodrigues, O. (1840).
Des lois géométriques qui régissent les déplacements d'un corps solide dans l'espace, et de la variation des coordonnées provenant de ces déplacements considérés indépendamment des causes qui peuvent les produire.
Journal de mathématiques pures et appliquées.

Rosten, E. and Drummond, T. (2006).
Machine learning for high-speed corner detection.
In *ECCV*.

Scharstein, D. and Szeliski, R. (2002).
A taxonomy and evaluation of dense two-frame stereo correspondence algorithms.
IJCV.

Schittkowski, K. (1985).
Nlqpl : A fortran-subroutine solving constrained nonlinear programming problems.
Annals of Operations Research.

Shanno, D. (1970).
Conditioning of quasi-newton methods for function minimization.
Mathematics of Computation.

Shi, J. and Tomasi, C. (1994).
Good features to track.
In *CVPR*.

Shoemake, K. (1985).
Animating rotation with quaternion curves.
SIGGRAPH.

Spivak, M. (1999).
A Comprehensive Introduction to Differential Geometry.
Publish or Perish.

Sturm and Maybank (1999).
On plane-based camera calibration : A general algorithm, singularities, applications.
In *CVPR*.

Sturm, J. F. (2001).
Using SeDuMi, a MATLAB toolbox for optimization over symmetric cones.

Thalin, G. (2011).
Deshaker v3.
http://www.guthspot.se/video/deshaker.htm.

Toh, K., Todd, M., and Tutuncu, R. (1999).
Sdpt3 — a matlab software package for semidefinite programming.

Optimization Methods and Software.

Triggs, B. (1997).
Autocalibration and the absolute quadric.
In *CVPR*.

Triggs, B. (1998).
Autocalibration from planar scenes.
In *ECCV*.

Triggs, B., Mclauchlan, P., Hartley, R., and Fitzgibbon, A. (2000).
Bundle adjustment – a modern synthesis.
Vision Algorithms : Theory and Practice.

Tsai, R. (1986).
An efficient and accurate camera calibration technique for 3D machine vision.
In *CVPR*.

Vandenberghe, L. and Boyd, S. (1996).
Semidefinite programming.
SIAM.

Vedaldi, A. and Fulkerson, B. (2008).
VLFeat : An open and portable library of computer vision algorithms.
http://www.vlfeat.org/.

Vogel, C. (1996).
Non-convergence of the l-curve regularization parameter selection method.
Inverse Problems.

Wahba, G. and Wold, S. (1975).
A completely automatic french curve : Fitting spline functions by cross validation.
Communications in Statistics.

Whaley, R. C., Petitet, A., and Dongarra, J. J. (2001).
Automated empirical optimization of software and the ATLAS project.
Parallel Computing.

Wilburn, B., Joshi, N., Vaish, V., Levoy, M., and Horowitz, M. (2004).
High speed video using a dense camera array.
In *CVPR*.

Yu, J. and McMillan, L. (2004).
General linear cameras.
In *ECCV*.

Zhang (1999).
Flexible camera calibration by viewing a plane from unknown orientations.
In *ICCV*.